

# Implicit stochastic gradient descent

Panos Toulis<sup>α</sup> and Edoardo M. Airolidi<sup>α</sup>

<sup>α</sup>Department of Statistics, Harvard University,  
Cambridge, MA, 02138, USA

October 6, 2015

## Abstract

Stochastic optimization procedures, such as stochastic gradient descent, have gained popularity for parameter estimation from large data sets. However, standard stochastic optimization procedures cannot effectively combine numerical stability with statistical and computational efficiency. Here, we introduce an *implicit* stochastic gradient descent procedure, the iterates of which are implicitly defined. Intuitively, implicit iterates *shrink* the standard iterates. The amount of shrinkage depends on the observed Fisher information matrix, which does not need to be explicitly computed in practice, thus increasing stability without increasing the computational burden. When combined with averaging, the proposed procedure achieves statistical efficiency as well. We derive non-asymptotic bounds and characterize the asymptotic distribution of implicit procedures. Our analysis also reveals the asymptotic variance of a number of existing procedures. We demonstrate implicit stochastic gradient descent by further developing theory for generalized linear models, Cox proportional hazards, and M-estimation problems, and by carrying out extensive experiments. Our results suggest that the implicit stochastic gradient descent procedure is poised to become the workhorse of estimation with large data sets.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Overview of main results</b>	<b>5</b>
2.1	Illustrative example . . . . .	7
2.2	Experimental evaluation . . . . .	8
2.3	Related work . . . . .	8
<b>3</b>	<b>Theory</b>	<b>10</b>
3.1	Non-asymptotic bounds . . . . .	11
3.2	Asymptotic variance and optimal learning rates . . . . .	12
3.2.1	Optimal learning rates . . . . .	13
3.3	Optimality with averaging . . . . .	16
3.4	Asymptotic normality . . . . .	17
3.5	Stability . . . . .	17
<b>4</b>	<b>Applications</b>	<b>19</b>
4.1	Efficient computation of implicit updates . . . . .	19
4.2	Generalized linear models . . . . .	19
4.3	Cox proportional hazards . . . . .	20
4.4	M-Estimation . . . . .	22
<b>5</b>	<b>Simulation and data analysis</b>	<b>23</b>
5.1	Validation of theory . . . . .	24
5.1.1	Asymptotic variance . . . . .	24
5.1.2	Asymptotic normality . . . . .	26
5.2	Validation of performance . . . . .	28
5.2.1	Experiments with <code>glm()</code> function . . . . .	28
5.2.2	Experiments with <code>biglm</code> . . . . .	29
5.2.3	Experiments with <code>glmnet</code> . . . . .	30
5.2.4	Cox proportional hazards . . . . .	31
5.2.5	M-estimation . . . . .	33
5.2.6	National Morbidity-Mortality Air Pollution study . . . . .	34
<b>6</b>	<b>Discussion</b>	<b>35</b>
<b>7</b>	<b>Conclusion</b>	<b>38</b>
<b>A</b>	<b>R code</b>	<b>1</b>
<b>B</b>	<b>Useful lemmas</b>	<b>1</b>

<b>C</b>	<b>Non-asymptotic analysis</b>	<b>5</b>
<b>D</b>	<b>Asymptotic analysis</b>	<b>7</b>
<b>E</b>	<b>Stability</b>	<b>15</b>
<b>F</b>	<b>Applications</b>	<b>16</b>
<b>G</b>	<b>Additional experiments</b>	<b>18</b>
G.1	Normality experiments with implicit SGD . . . . .	18
G.2	Poisson regression . . . . .	18
G.3	Experiments with <code>glmnet</code> . . . . .	20
G.4	Support vector machines . . . . .	20
G.5	Experiments with machine learning algorithms . . . . .	21
G.5.1	Averaged explicit SGD . . . . .	22
G.5.2	Prox-SVRG and Prox-SAG . . . . .	23

# 1 Introduction

Parameter estimation by optimization of an objective function, such as maximum likelihood and maximum a-posteriori, is a fundamental idea in statistics and machine learning (Fisher, 1922; Lehmann and Casella, 1998; Hastie et al., 2011). However, widely used optimization-based estimation procedures, such as Fisher scoring, the EM algorithm or iteratively reweighted least squares (Fisher, 1925; Dempster et al., 1977; Green, 1984), do not scale to modern data sets with millions of data points and hundreds or thousands of parameters (National Research Council, 2013). In this paper, we introduce and further develop iterative estimation procedures based on *stochastic gradient descent* (SGD) optimization, rooted in the early statistics literature on stochastic approximations (Robbins and Monro, 1951), which are computationally efficient and lead to estimates with good statistical properties.

We consider the problem of estimating the true vector of parameters  $\theta_\star \in \mathbf{R}^p$  of a model that is assumed to produce i.i.d. data points  $(X_i, Y_i)$ , for  $i = 1, 2, \dots, N$ . Conditional on covariates  $X_i \in \mathbf{R}^p$ , the outcome  $Y_i \in \mathbf{R}^d$  is distributed according to known density  $f(Y_i; X_i, \theta_\star)$ . The *expected Fisher information matrix* is  $\mathcal{I}(\theta_\star) \stackrel{\text{def}}{=} \mathbb{E}(\nabla \log f(Y; X, \theta_\star) \nabla \log f(Y; X, \theta_\star)^\top)$ , where  $(X, Y)$  denotes a data point. Properties of the Fisher information matrix are very important for the stability and efficiency of SGD procedures, as we show in Sections 3 and 3.5. Given a sample of  $N$  i.i.d. data points the estimation problem usually

reduces to optimization, for example, finding the maximum likelihood estimate (MLE), defined as  $\theta_N^{\text{mle}} = \arg \max_{\theta} \sum_{i=1}^N \log f(Y_i; X_i, \theta)$ .<sup>1</sup>

Traditional estimation procedures have a running time complexity that ranges between  $O(Np^{1+\epsilon})$  and  $O(Np^{2+\epsilon})$  in best cases and worst cases, respectively. Newton-Raphson, for instance, converges linearly to MLE (Kelley, 1999), however, matrix inversions and likelihood computations on the entire data set yield an algorithm with  $O(Np^{2+\epsilon})$  complexity, per iteration, which makes it unsuitable for large data sets. Fisher scoring, a variant of Newton-Raphson, is generally more stable but has similar computational properties (Lange, 2010). Quasi-Newton (QN) procedures are a powerful alternative and routinely used in practice<sup>2</sup> because they have  $O(Np^2)$  complexity per-iteration, or  $O(Np^{1+\epsilon})$  in certain favorable cases. (Hennig and Kiefel, 2013). Other general estimation algorithms, such as EM (Dempster et al., 1977) or iteratively reweighted least squares (Green, 1984), involve computations (e.g. matrix inversions or maximizations between iterations) that are significantly more expensive than QN procedures.

In contrast, estimation with large data sets requires a running time complexity that is roughly  $O(Np^{1-\epsilon})$ , i.e., linear in data size  $N$  but sublinear in parameter dimension  $p$ . The first requirement on  $N$  is rather unavoidable because all data points carry information by the i.i.d. assumption, and thus all need to be considered by any iterative estimation procedure. Thus, sublinearity in  $p$  is crucial. Such computational requirements have recently sparked interest in stochastic optimization procedures, especially those only working with *first-order* information, i.e., gradients.

Stochastic optimization procedures of this kind are rooted in stochastic approximation methods (Robbins and Monro, 1951), where, interestingly, the estimation problem is formulated not as an optimization problem, but as a *characteristic equation*. In particular, if  $N$  is finite the characteristic equation is

$$\mathbb{E}(\nabla \log f(Y; X, \theta_N^{\text{mle}})) = 0, \quad (1)$$

where the expectation is over the *empirical* distribution of  $(X, Y)$  on the finite data set. If  $N$  is infinite – a situation known as “stream of data” – the characteristic equation is

$$\mathbb{E}(\nabla \log f(Y; X, \theta_{\star}) | X) = 0, \quad (2)$$

where the expectation is over the true conditional distribution of outcome  $Y$  given covariate  $X$ . Given a characteristic equation, stochastic approximations are pro-

---

<sup>1</sup>If a prior  $\pi(\theta)$ , also known as regularization, is assumed on the model parameters  $\theta$  then the estimation problem reduces to finding the maximum a-posteriori estimate (MAP), defined as  $\theta_N^{\text{map}} = \arg \max_{\theta} \sum_{i=1}^N \log f(Y_i; X_i, \theta) + \log \pi(\theta)$ .

<sup>2</sup>For example, most implemented algorithms in R’s `optim()` function are Quasi-Newton.

cedures that iteratively approximate its solution (Ljung et al., 1992; Benveniste et al., 1990), i.e., they approximate  $\theta_N^{\text{mle}}$  in Eq. (1) and  $\theta_*$  in Eq. (2).<sup>3</sup>

A popular stochastic approximation procedure for estimation with large data sets is *stochastic gradient descent* (SGD), defined for  $n = 1, 2, \dots$ , as

$$\theta_n^{\text{sgd}} = \theta_{n-1}^{\text{sgd}} + \gamma_n C_n \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{sgd}}), \quad (3)$$

where  $\gamma_n > 0$  is the *learning rate sequence*, typically defined as  $\gamma_n = \gamma_1 n^{-\gamma}$ ,  $\gamma_1 > 0$  is the *learning rate parameter*,  $\gamma \in (1/2, 1]$ , and  $C_n$  are  $p \times p$  positive-definite matrices, also known as condition matrices.

When  $N$  is finite the data point  $(X_n, Y_n)$  in SGD (3) is a random sample with replacement from the finite data set. When  $N$  is infinite the data point  $(X_n, Y_n)$  is simply the  $n$ th data point in the data stream. For the rest of this paper we assume infinite  $N$  because it is the natural setting for stochastic approximations. This assumption is without loss of generality because all theoretical results for the infinite  $N$  case can be applied to finite  $N$ , where instead of estimating  $\theta_*$  we estimate  $\theta_N^{\text{mle}}$  (or MAP if there is regularization).

From a computational perspective, SGD (3) is appealing because it avoids expensive matrix inversions, as in Newton-Raphson, and the log-likelihood is evaluated at a single data point  $(X_n, Y_n)$  and not on the entire data set. From a theoretical perspective, SGD (3) is essentially a stochastic approximation procedure, and thus converges, under suitable conditions, to  $\theta_\infty^{\text{sgd}}$  where  $\mathbb{E}(\log f(Y; X, \theta_\infty^{\text{sgd}}) | X) = 0$ . This condition satisfies both Eqs. (1) and (2), implying that SGD can be used on both finite and infinite data sets.

The main contribution of this paper is to provide a formal analysis and the statistical intuition behind *implicit* stochastic gradient descent, or *implicit SGD* for short, defined for  $n = 1, 2, \dots$ , as follows:

$$\theta_n^{\text{im}} = \theta_{n-1}^{\text{im}} + \gamma_n C_n \nabla \log f(Y_n; X_n, \theta_n^{\text{im}}), \quad (4)$$

where  $\gamma_n, C_n$  are defined as in standard SGD (3). To distinguish the two procedures, we will refer to SGD (3) as SGD with *explicit updates*, or *explicit SGD* for short, because the next iterate  $\theta_n^{\text{sgd}}$  can be immediately computed given  $\theta_{n-1}^{\text{sgd}}$  and the data point  $(X_n, Y_n)$ . In contrast, the update in Eq. (4) is *implicit* because the next iterate  $\theta_n^{\text{im}}$  appears on both sides of the equation.

## 2 Overview of main results

Our first set of results relies on a non-asymptotic analysis of iterates  $\theta_n^{\text{im}}$  of implicit SGD (4). In particular, we derive upper bounds for the mean-squared

---

<sup>3</sup>The characteristic equations can have a set of multiple solutions. In this case, stochastic approximations converge to a point in that set, but the exact point to which they converge depends on initial conditions (Borkar, 2008).

errors  $\mathbb{E}(\|\theta_n^{\text{im}} - \theta_\star\|^2)$ . Compared to non-asymptotic analyses of explicit SGD (Benveniste et al., 1990; Ljung et al., 1992; Moulines and Bach, 2011), implicit SGD is particularly robust to misspecification of the learning rate with respect to the problem characteristics (e.g., convexity). In contrast, in explicit SGD the initial conditions can be amplified arbitrarily if learning rates are misspecified.

Our second set of results relies on an asymptotic analysis of  $\theta_n^{\text{im}}$ . In particular, we show that the asymptotic variance of  $\theta_n^{\text{im}}$  is identical to the variance of  $\theta_n^{\text{sgd}}$ . Both procedures lose statistical information compared, for example, to MLE  $\theta_n^{\text{mle}}$  computed over  $n$  data points. However, this information loss can be quantified exactly (Theorem 3.2), which can be leveraged to design optimal learning rates (Eq. 9). Surprisingly, this information loss can be avoided by simply averaging the iterates  $\theta_n^{\text{im}}$  (Theorem 3.3). This result matches similar results on averaging of explicit SGD procedures, first given by Ruppert (1988); Bather (1989); Polyak and Juditsky (1992b). Additionally, under typical Lindeberg conditions, we show that  $\theta_n^{\text{im}}$  is asymptotically normal with known asymptotic variance, which can be leveraged for producing standard errors for implicit SGD estimates.

The combined results from the non-asymptotic and asymptotic analyses show that explicit SGD procedures cannot easily combine numerical stability with statistical efficiency (Section 3.5). In rough terms, for stability we need  $\lambda_{\max}\gamma_1 < 1$ , where  $\gamma_1$  is the learning rate parameter, and  $\lambda_{\max}$  is the maximum eigenvalue of the Fisher information  $\mathcal{I}(\theta_\star)$ . For statistical efficiency we need  $\lambda_{\min}\gamma_1 > 1/2$ , where  $\lambda_{\min}$  is the minimum eigenvalue of  $\mathcal{I}(\theta_\star)$ . Thus, we need  $\lambda_{\max} < 2\lambda_{\min}$  to achieve both stability and efficiency. This condition depends on the *condition number* of  $\mathcal{I}(\theta_\star)$  and is hard to satisfy in large data sets with high-dimensional parameters. In stark contrast, the stability condition is eliminated in implicit SGD because, effectively, any learning rate parameter  $\gamma_1$  can yield a stable procedure (Theorem 3.1). With stability issues fixed, one has more freedom to select larger learning rates, with or without averaging, in order to speed up convergence and increase statistical efficiency, which sums up the benefits of implicit over explicit SGD.

Our third set of results relies on practical applications of implicit SGD on a wide family of statistical models. In particular, we devise a new algorithm for fast calculation of iterates  $\theta_n^{\text{im}}$  (Algorithm 1) by solving efficiently the  $p$ -dimensional fixed-point equation (4) in the definition of implicit SGD. Algorithms 2, 4, and 5 are variants that specialize Algorithm 1 to generalized linear models, Cox proportional hazards, and M-estimation problems, respectively. Outside of such models, we show how implicit updates can be approximately implemented (Sections 5.2.4 & 6), and show empirical evidence that this approximation works well on Cox proportional hazards models (Section 5.2.4), and a big SVM model (Section G.4).

## 2.1 Illustrative example

We motivate our main results through a simple estimation problem. Let  $\theta_\star \in \mathbf{R}$  be the true parameter of a normal model with i.i.d. observations  $Y_i|X_i \sim \mathcal{N}(X_i\theta_\star, \sigma^2)$ , where the variance  $\sigma^2$  is assumed known for simplicity. The log-likelihood is  $\log f(Y_i; X_i, \theta) = -\frac{1}{2\sigma^2}(Y_i - X_i\theta)^2$ , and the score function (i.e., gradient of log-likelihood) is given by  $\nabla \log f(Y_i; X_i, \theta) = \frac{1}{\sigma^2}(Y_i - X_i\theta)X_i$ . Let  $X_i$  be distributed according to some unknown distribution with bounded second. Assume  $\gamma_n = \gamma_1/n$ , for some  $\gamma_1 > 0$  as the learning rate, and an initial condition  $\theta_0$ . Then, the explicit SGD procedure (3) is

$$\begin{aligned}\theta_n^{\text{sgd}} &= \theta_{n-1}^{\text{sgd}} + \gamma_n(Y_n - \theta_{n-1}^{\text{sgd}}X_n)X_n \Rightarrow \\ \theta_n^{\text{sgd}} &= (1 - \gamma_n X_n^2)\theta_{n-1}^{\text{sgd}} + \gamma_n Y_n X_n.\end{aligned}\tag{5}$$

Procedure (5) is the least mean squares filter (LMS) in signal processing, also known as the Widrow-Hoff algorithm (Widrow and Hoff, 1960). The implicit SGD procedure can be derived in closed form in this problem using update (4) as

$$\begin{aligned}\theta_n^{\text{im}} &= \theta_{n-1}^{\text{im}} + \gamma_n(Y_n - X_n\theta_{n-1}^{\text{im}})X_n \Rightarrow \\ \theta_n^{\text{im}} &= \frac{1}{1 + \gamma_n X_n^2}\theta_{n-1}^{\text{im}} + \frac{\gamma_n}{1 + \gamma_n X_n^2}Y_n X_n.\end{aligned}\tag{6}$$

Procedure (6) is known as the normalized least mean squares filter (NLMS) in signal processing (Nagumo and Noda, 1967).

From Eq. (5) we see that it is crucial for explicit SGD to have a well-specified learning rate parameter  $\gamma_1$ . For instance, if  $\gamma_1 X_n^2 \gg 1$  then  $\theta_n^{\text{sgd}}$  will diverge to a value at the order of  $2^n/\sqrt{\gamma_1}$ , before converging to the true value (see Section 3.5, Lemma 3.1). In contrast, implicit SGD is more stable to misspecification of the learning rate parameter  $\gamma_1$ . For example, a very large  $\gamma_1$  will not cause divergence as in explicit SGD, but it will simply put more weight on the  $n$ th observation  $Y_n X_n$  than the previous iterate  $\theta_{n-1}^{\text{im}}$ . Assuming for simplicity  $\theta_{n-1}^{\text{sgd}} = \theta_{n-1}^{\text{im}} = 0$ , it also holds  $\theta_n^{\text{im}} = \frac{1}{1 + \gamma_n X_n^2}\theta_n^{\text{sgd}}$ , showing that implicit SGD iterates are shrunk versions of explicit ones (see also Section 6).

Let  $v^2 \stackrel{\text{def}}{=} \mathbb{E}(X^2)$ , then according to Theorem 3.2 the asymptotic variance of the explicit iterate  $\theta_n^{\text{sgd}}$  (and the implicit  $\theta_n^{\text{im}}$ ) satisfies  $n\text{Var}(\theta_n^{\text{sgd}}) \rightarrow \gamma_1^2 \sigma^2 v^2 / (2\gamma_1 v^2 - 1)$  if  $2\gamma_1 v^2 - 1 > 0$ . Since  $\gamma_1^2 / (2\gamma_1 v^2 - 1) \geq 1/v^2$ , it is best to set  $\gamma_1 = 1/v^2$ . In this case  $n\text{Var}(\theta_n^{\text{sgd}}) \rightarrow \sigma^2/v^2$ . Explicit SGD can thus be optimal by setting  $\gamma_n = (\sum_{i=1}^n X_i^2)^{-1}$ , which implies that  $\theta_n^{\text{sgd}} = \sum_{i=1}^n Y_i X_i / \sum_{i=1}^n X_i^2$ , i.e., it is the classical OLS estimator. The implicit SGD estimator  $\theta_n^{\text{im}}$  (6) inherits the efficiency properties of  $\theta_n^{\text{sgd}}$ , with the added benefit of being stable over a wide range of learning rates  $\gamma_1$ . Overall, implicit SGD is a superior form of SGD.

## 2.2 Experimental evaluation

Our experiments are split into two sets. First, in Section 5.1 we perform experiments to validate our theoretical findings. In particular we test our results on asymptotic variance of first-order or second-order SGD procedures (Section 3.2.1, Theorem 3.2), and our results on asymptotic normality (Theorem 3.4).

Second, in Section 5.2, we carry out an extensive set of experiments on real and simulated data to compare the performance of implicit SGD against optimal deterministic optimization procedures. In particular, we compare implicit SGD against deterministic procedures on statistical models presented in Sections 4.2-4.4. Subsequently, we focus on generalized linear models and compare implicit SGD against Fisher scoring (using the R `glm()` function) on various generalized linear models, and against a popular alternative to scale Fisher scoring to large data sets (using the R `biglm` package). We also compare implicit SGD against the elastic net by Friedman et al. (2010) for sparse regularized estimation (using the R `glmnet` package). Finally, we work on a variant of generalized linear models, where we re-analyze a large data set from the National Morbidity-Mortality Air Pollution study (Samet et al., 2000; Dominici et al., 2002), and compare against recently published methods that were specifically designed for that task.

Additional experiments are presented in the Appendix. In particular, we perform additional experiments to compare implicit SGD with known adaptive or proximal stochastic optimization methods, including AdaGrad (Duchi et al., 2011a), Prox-SVRG (Xiao and Zhang, 2014), and Prox-SAG (Schmidt et al., 2013), on popular machine learning tasks. Overall, our results provide strong evidence that the family of implicit SGD procedures provides a superior form of stochastic gradient descent, with comparable statistical efficiency to explicit SGD, but with much improved stability. Combined with results on asymptotic variance and normality, implicit SGD emerges as a principled estimation method that can become the workhorse of statistical estimation with large data sets in statistical practice.

## 2.3 Related work

Historically, the duo of explicit-implicit updates originate from the numerical methods introduced by Euler (ca. 1770) for approximating solutions of ordinary differential equations (Hoffman and Frankel, 2001). The explicit SGD procedure was first proposed by Sakrison (1965) as a recursive statistical estimation method and it is theoretically based on the stochastic approximation method of Robbins and Monro (1951). Statistical estimation with explicit SGD is a straightforward generalization of Sakrison’s method and has recently attracted attention in the machine learning community as a fast learning method for large-scale problems



(Zhang, 2004; Bottou, 2010; Toulis and Airoldi, 2015). Applications of explicit SGD procedures in massive data problems can be found in many diverse areas such as large-scale machine learning (Zhang, 2004; Bottou, 2010), online EM algorithm (Cappé and Moulines, 2009; Balakrishnan et al., 2014), image analysis (Lin et al., 2011), deep learning (Dean et al., 2012; Erhan et al., 2010) and MCMC sampling (Welling and Teh, 2011).

The implicit SGD procedure is less known and not well-understood. In optimization, implicit methods have recently attracted attention under the guise of proximal methods, or mirror-descent methods (Nemirovski, 1983; Beck and Teboulle, 2003). In fact, the implicit SGD update (4) can be expressed as a proximal update as the solution of

$$\theta_n^{\text{im}} = \arg \max_{\theta} \left\{ -\frac{1}{2} \|\theta - \theta_{n-1}^{\text{im}}\|^2 + \gamma_n \log f(Y_n; X_n, \theta) \right\}. \quad (7)$$

From a Bayesian perspective,  $\theta_n^{\text{im}}$  of the implicit procedure is the posterior mode of a model with the standard multivariate normal  $\mathcal{N}(\theta_{n-1}^{\text{im}}, \gamma_n I)$  as the prior, and  $\log f(Y_n; X_n, \theta)$  as the log-likelihood of  $\theta$  for observation  $(X_n, Y_n)$ . Arguably, the normalized least mean squares (NLMS) filter (Nagumo and Noda, 1967), introduced in Eq. (6), was the first statistical model that used an implicit update as in Equation (4) and was shown to be consistent and robust to input noise (Slock, 1993). From an optimization perspective, update (7) corresponds to a stochastic version of the proximal point algorithm by Rockafellar (1976) which has been generalized through the idea of splitting algorithms (Lions and Mercier, 1979; Beck and Teboulle, 2009; Singer and Duchi, 2009; Duchi et al., 2011b); see, also, the comprehensive review of proximal methods in optimization by Parikh and Boyd (2013). Additional intuition of implicit methods have been given by Krakowski et al. (2007) and Nemirovski et al. (2009) who have argued that proximal methods can fit better in the geometry of the parameter space. Bertsekas (2011) derived an asymptotic rate for an implicit procedure (4) on a fixed data set and compared convergence rates between randomly sampling data  $(X_n, Y_n)$  and simply cycling through them. Toulis et al. (2014) derived the asymptotic variance of  $\theta_n^{\text{im}}$  as estimator of  $\theta_*$  in the family of generalized linear models, and provided an algorithm to efficiently compute the update (4). Rosasco et al. (2014) derived non-asymptotic bounds and convergence for a stochastic proximal gradient algorithm, which is a forward-backward procedure that first makes as stochastic explicit update (3), and then a deterministic implicit update.

In the online learning literature, “regret analyses” of implicit methods have been given by Kivinen et al. (2006) and Kulis and Bartlett (2010); Schuurmans and Caelli (2007) have further applied implicit methods on learning with kernels. Furthermore, the proximal update (7) is related to the importance weight updates

proposed by [Karampatziakis and Langford \(2010\)](#), but the two updates have important differences ([Karampatziakis and Langford, 2010](#), Section 5).

Two recent stochastic proximal methods are **Prox-SVRG** ([Xiao and Zhang, 2014](#)) and **Prox-SAG** ([Schmidt et al., 2013](#), Section 6). Working in a finite data set, the main idea in both methods is to periodically compute an estimate of the full gradient averaged over all data points to reduce the variance of stochastic gradients. This periodic update is also controlled by additional hyperparameters, whereas **Prox-SAG** typically requires storage of the full gradient at every iteration. We compare those methods with implicit SGD in Appendix G.

### 3 Theory

The norm  $\|\cdot\|$  denotes the  $L_2$  norm. If a positive scalar sequence  $a_n$  is nonincreasing and  $a_n \rightarrow 0$ , we write  $a_n \downarrow 0$ . For two positive scalar sequences  $a_n, b_n$ , equation  $b_n = O(a_n)$  denotes that  $b_n$  is bounded above by  $a_n$ , i.e., there exists a fixed  $c > 0$  such that  $b_n \leq ca_n$ , for all  $n$ . Furthermore,  $b_n = o(a_n)$  denotes that  $b_n/a_n \rightarrow 0$ . Similarly, for a sequence of vectors (or matrices)  $X_n$ , we write  $X_n = O(a_n)$  if there is a fixed  $c' > 0$  such that  $\|X_n\| \leq c'a_n$ , and  $X_n = o(a_n)$  if  $\|X_n\|/a_n \rightarrow 0$ . For two matrices  $A, B$   $A \prec B$  denotes that  $B - A$  is positive-definite. The set of eigenvalues of a matrix  $A$  is denoted by  $\text{eig}(A)$ ; for example,  $A \succ 0$  if and only if  $\lambda > 0$  for every  $\lambda \in \text{eig}(A)$ .

Every theoretical result of this section is stated under a combination of the following assumptions.

**Assumption 3.1.** *The explicit SGD procedure (3) and the implicit SGD procedure (4) operate under a combination of the following assumptions.*

- (a) *The learning rate sequence  $\{\gamma_n\}$  is defined as  $\gamma_n = \gamma_1 n^{-\gamma}$ , where  $\gamma_1 > 0$  is the learning parameter, and  $\gamma \in (0.5, 1]$ .*
- (b) *For the log-likelihood  $\log f(Y; X, \theta)$  there exists function  $\ell$  such that  $\log f(Y; X, \theta) \equiv \ell(X^\top \theta; Y)$ , which depends on  $\theta$  only through the natural parameter  $X^\top \theta$ .*
- (c) *Function  $\ell$  is concave, twice differentiable almost-surely with respect to natural parameter  $X^\top \theta$  and Lipschitz continuous with constant  $L_0$ .*
- (d) *The observed Fisher information matrix  $\hat{\mathcal{I}}_n(\theta) \triangleq -\nabla^2 \ell(X_n^\top \theta; Y_n)$  satisfies, almost-surely,  $\hat{\mathcal{I}}_n(\theta) \succeq \underline{\lambda}_f I$ , where  $\underline{\lambda}_f > 0$ . The Fisher information matrix  $\mathcal{I}(\theta_\star) \stackrel{\text{def}}{=} \mathbb{E} \left( \hat{\mathcal{I}}_n(\theta_\star) \right)$  has maximum eigenvalue  $\overline{\lambda}_f < \infty$ . Typical regularity conditions hold ([Lehmann and Casella, 1998](#), Theorem 5.1, p.463).*
- (e) *Every condition matrix  $C_n$  is a fixed positive-definite matrix, such that  $C_n = C + O(\gamma_n)$ , where  $C \succ 0$  and symmetric, and  $C$  commutes with  $\mathcal{I}(\theta_\star)$ . For every  $C_n$ ,  $\min \text{eig}(C_n) \geq \underline{\lambda}_c > 0$ , and  $\max \text{eig}(C_n) \leq \overline{\lambda}_c < \infty$ .*

(f) Let  $\Xi_n \stackrel{\text{def}}{=} \mathbb{E}(\nabla \log f(Y_n; X_n, \theta_\star) \nabla \log f(Y_n; X_n, \theta_\star)^\top | \mathcal{F}_{n-1})$ , then  $\|\Xi_n - \Xi\| = O(1)$  for all  $n$ , and  $\|\Xi_n - \Xi\| \rightarrow 0$ , for a symmetric positive-definite  $\Xi$ . Let  $\sigma_{n,s}^2 \stackrel{\text{def}}{=} \mathbb{E}(\mathbb{I}_{\|\xi_n(\theta_\star)\|^2 \geq s/\gamma_n} \|\xi_n(\theta_\star)\|^2)$ , then for all  $s > 0$ ,  $\sum_{i=1}^n \sigma_{i,s}^2 = o(n)$  if  $\gamma = 1$ , and  $\sigma_{n,s}^2 = o(1)$  otherwise.

*Remarks.* Assumption 3.1(a) is typical in stochastic approximation as it implies  $\sum_i \gamma_i = \infty$  and  $\sum_i \gamma_i^2 < \infty$ , which were the original conditions given by Robbins and Monro (1951). Assumption 3.1(b) narrows our focus to models for which, conditional on covariate  $X$ , the likelihood depends on parameters  $\theta$  through the linear combination  $X^\top \theta$ . This family of models is large and includes generalized linear models, Cox proportional hazards models, and M-estimation. Furthermore, in Section 6 we discuss an idea that allows application of implicit SGD on a wider family of models, and thus can significantly relax Assumption 3.1(b). Assumption 3.1(d) is equivalent to assuming strong convexity for the negative log-likelihood. Such assumptions are typical for proving convergence in probability. The assumption on the observed Fisher information is less standard. Intuitively, this assumption posits that a minimum of statistical information is received from any data point, at least for certain model parameters. Making this assumption allows us to forgo boundedness assumptions on the errors of stochastic gradients that were originally used by Robbins and Monro (1951) and have since been standard in analysis of explicit SGD. Finally, Assumption 3.1(f) posits the typical Lindeberg conditions that are necessary to invoke the central limit theorem and prove asymptotic optimality; this assumption follows the conditions defined by Fabian (1968a) for the normality of explicit SGD procedures.

### 3.1 Non-asymptotic bounds

In this section we derive non-asymptotic upper-bounds for the errors  $\|\theta_n^{\text{im}} - \theta_\star\|^2$  in expectation.

**Theorem 3.1.** Let  $\delta_n \triangleq \mathbb{E}(\|\theta_n^{\text{im}} - \theta_\star\|^2)$  and  $\kappa \triangleq 1 + \gamma_1 \underline{\lambda}_c \underline{\lambda}_f \mu_0$ , where  $\mu_0 \in [1/(1 + \gamma_1 \underline{\lambda}_f \underline{\lambda}_c (p-1)), 1]$ . Suppose that Assumptions 3.1(a),(b),(c), (d) and (e) hold. Then, there exists constant  $n_0$  such that,

$$\delta_n \leq \frac{8L_0^2 \bar{\lambda}_c^2 \gamma_1 \kappa}{\underline{\lambda}_c \underline{\lambda}_f \mu_0} n^{-\gamma} + \exp(-\log \kappa \cdot n^{1-\gamma}) [\delta_0 + \kappa^{n_0} \Gamma^2],$$

where  $\Gamma^2 = 4L_0^2 \bar{\lambda}_c^2 \sum_i \gamma_i^2 < \infty$ , and  $n_0$  is defined in Corollary B.1.

Not surprisingly, implicit SGD (4) matches the asymptotic rate of explicit SGD (3). In particular, the iterates  $\theta_n^{\text{im}}$  have squared error  $O(n^{-\gamma})$ , as seen in Theorem 3.1, which is identical to the squared error of the explicit iterates  $\theta_n^{\text{sgd}}$

(Benveniste et al., 1990, Theorem 22, p.244). Furthermore, we will show in the following section that both iterates have the same asymptotic efficiency when viewed as estimators of  $\theta_*$ .

However, the critical advantage of implicit SGD – more generally of implicit procedures – is their robustness to initial conditions and excess noise. This can be seen in Theorem 3.1 where the implicit procedure discounts the initial conditions  $\mathbb{E}(\|\theta_0^{\text{im}} - \theta_*\|^2)$  at an exponential rate through the term  $\exp(-\log(1 + 2\gamma_1 \underline{\lambda}_c \underline{\lambda}_f) n^{1-\gamma})$ , where  $\gamma_1$  is the learning rate parameter, and  $\underline{\lambda}_c, \underline{\lambda}_f$  are minimum eigenvalues of the condition and the Fisher information matrices, respectively. Importantly, the discounting of initial conditions happens regardless of the specification of the learning rate. In fact, large values of  $\gamma_1$  can lead to faster discounting, and thus possibly to faster convergence, however at the expense of increased variance. The implicit iterates are therefore *unconditionally stable*, i.e., virtually any specification of the learning rate will lead to a stable discounting of the initial conditions.

In stark contrast, explicit SGD is known to be very sensitive to the learning rate, and can numerically diverge if the rate is misspecified. For example, Moulines and Bach (2011, Theorem 1) showed that there exists a term  $\exp(L^2 \gamma_1^2 n^{1-2\gamma})$ , where  $L$  is a Lipschitz constant for the gradient of the log-likelihood, amplifying the initial conditions  $\mathbb{E}(\|\theta_0^{\text{sgd}} - \theta_*\|^2)$  of explicit SGD, which can be catastrophic if the learning rate parameter  $\gamma_1$  is misspecified.<sup>4</sup> Thus, although implicit and explicit SGD have identical asymptotic performance, they are crucially different in their stability properties. This is confirmed in the experiments of Section 5.

### 3.2 Asymptotic variance and optimal learning rates

In the previous section we showed that  $\theta_n^{\text{im}} \rightarrow \theta_*$  in quadratic mean, i.e., the implicit SGD iterates converge to the true model parameters  $\theta_*$ , similar to classical results for the explicit SGD iterates  $\theta_n^{\text{sgd}}$ . Thus,  $\theta_n^{\text{im}}$  and  $\theta_n^{\text{sgd}}$  are consistent estimators of  $\theta_*$ . In the following theorem we show that both SGD estimators have the same asymptotic variance.

**Theorem 3.2.** *Consider SGD procedures (3) and (4), and suppose that Assumptions 3.1(a),(c),(d),(e) hold, where  $\gamma = 1$ . The asymptotic variance of the explicit*

---

<sup>4</sup>The Lipschitz conditions are different in the two works, however this does not affect our conclusion. Our result remains effectively unchanged if we assume Lipschitz continuity of the gradient  $\nabla \ell$  instead of the log-likelihood  $\ell$ , similar to Moulines and Bach (2011); see comment after proof of Theorem 3.1.

SGD estimator (3) satisfies

$$n\text{Var}(\theta_n^{\text{sgd}}) \rightarrow \gamma_1^2 (2\gamma_1 C\mathcal{I}(\theta_\star) - I)^{-1} C\mathcal{I}(\theta_\star)C.$$

The asymptotic variance of the implicit SGD estimator (4) satisfies

$$n\text{Var}(\theta_n^{\text{im}}) \rightarrow \gamma_1^2 (2\gamma_1 C\mathcal{I}(\theta_\star) - I)^{-1} C\mathcal{I}(\theta_\star)C.$$

*Remarks.* Although the implicit SGD estimator  $\theta_n^{\text{im}}$  is significantly more stable than the explicit estimator  $\theta_n^{\text{sgd}}$  (Theorem 3.1), both estimators have the same asymptotic efficiency in the limit according to Theorem 3.2. This implies that implicit SGD is a superior form of SGD, and should be preferred when the calculation of implicit updates (4) is computationally feasible. In Section 4 we show that this is possible in a large family of statistical models, and illustrate with several numerical experiments in Section 5.1.

Asymptotic variance results in stochastic approximation similar to Theorem 3.2 were first obtained by Chung (1954), Sacks (1958), and followed by Fabian (1968b), Polyak and Tsykin (1979), and several other authors (see also Ljung et al., 1992, Parts I, II). Our work is different in two important aspects. First, our asymptotic variance result includes implicit SGD, which is a stochastic approximation procedure with implicitly defined updates, whereas other works consider only explicit stochastic approximation procedures. Second, in our setting we estimate recursively the true parameters  $\theta_\star$  of a statistical model, and thus we can exploit typical regularity conditions (Assumption 3.1(d)) to derive the asymptotic variance of  $\theta_n^{\text{im}}$  (and  $\theta_n^{\text{sgd}}$ ) in a simplified closed-form; for example, under typical regularity conditions (see also Assumption 3.1(d)),  $\text{Var}(\nabla \log f(Y; X, \theta_\star)) = \mathcal{I}(\theta_\star)$ , which is used in the proof of Theorem 3.2. We illustrate the asymptotic variance results of Theorem 3.2 in experiments of Section 5.1.1.

### 3.2.1 Optimal learning rates

Crucially, the asymptotic variance formula of Theorem 3.2 depends on the limit of the sequence  $C_n$  used in the SGD procedures (3) and (4). We distinguish two classes of procedures, one where  $C_n = I$  trivially, known as *first-order procedures*, and a second case where  $C_n$  is not trivial, known as *second-order procedures*.

In first-order procedures,  $C_n = I$ , i.e., only gradients are used in the SGD procedures. Inevitably, no matter how we set the learning rate parameter  $\gamma_1$ , first-order SGD procedures will lose statistical efficiency. We can immediately verify this by comparing the asymptotic variance in Theorem 3.2 with the asymptotic variance of the maximum likelihood estimator (MLE), denoted by  $\theta_N^{\text{mle}}$ , on a data set with  $N$  data points  $\{(X_n, Y_n)\}$ ,  $n = 1, 2, \dots, N$ . Under regularity conditions, the MLE is the asymptotically optimal unbiased estimator and

$N\text{Var}(\theta_N^{\text{mle}} - \theta_\star) \rightarrow \mathcal{I}(\theta_\star)^{-1}$ . By Theorem 3.2 and convergence of implicit SGD, it holds  $N\text{Var}(\theta_n^{\text{im}} - \theta_\star) \rightarrow \gamma_1^2(2\gamma_1\mathcal{I}(\theta_\star) - I)^{-1}\mathcal{I}(\theta_\star)$ , which also holds for  $\theta_n^{\text{sgd}}$ . For any  $\gamma_1 > 0$  we have,

$$\gamma_1^2(2\gamma_1\mathcal{I}(\theta_\star) - I)^{-1}\mathcal{I}(\theta_\star) \succeq \mathcal{I}(\theta_\star)^{-1}. \quad (8)$$

Therefore, both SGD estimators lose information and this loss can be quantified exactly by Ineq. (8). This inequality can also be leveraged to find the optimal choice for  $\gamma_1$  given an appropriate objective. As demonstrated in the experiments in Section 5, this often suffices to achieve estimates that are comparable with MLE in statistical efficiency but with substantial computational gains. Assuming distinct eigenvalues  $\lambda_i$  for the matrix  $\mathcal{I}(\theta_\star)$ , the eigenvalues of the variance matrix of both SGD estimators are equal to  $\gamma_1^2\lambda_i/(2\gamma_1\lambda_i - 1)$ , by Theorem 3.2. Therefore, one reasonable way to set the parameter  $\gamma_1$  is to set it equal to  $\gamma_1^\star$ , defined as

$$\gamma_1^\star = \arg \min_{x > 1/2\lambda_f} \sum_i x^2\lambda_i/(2x\lambda_i - 1). \quad (9)$$

Eq. (9) is under the constraint  $x > 1/(2\lambda_f)$ , where  $\lambda_f = \min\{\lambda_i\}$ , because Theorem 3.2 requires  $2\gamma_1\mathcal{I}(\theta_\star) - I$  to be positive-definite, and thus  $(2x\lambda_i - 1)$  needs to be positive for every  $\lambda_i$ .

Of course, the eigenvalues  $\lambda_i$ 's are unknown in practice and need to be estimated from the data. This problem has received significant attention recently and several methods exist (see Karoui, 2008, and references within). We will use Eq. (9) extensively in our experiments (Section 5) in order to tune the SGD procedures. However, we note that in first-order SGD procedures, knowing the eigenvalues  $\lambda_i$  of  $\mathcal{I}(\theta_\star)$  does not necessarily achieve statistical efficiency because of the spectral gap of  $\mathcal{I}(\theta_\star)$ , i.e., the ratio between its maximum eigenvalue  $\bar{\lambda}_f$  and minimum eigenvalue  $\lambda_f$ ; for instance, if  $\lambda_f = \bar{\lambda}_f$ , then the choice of learning rate parameter (9) leads to statistically efficient first-order SGD procedures. However, this case is not typical in practice, especially in many dimensions.

In *second-order* procedures, we assume non-trivial condition matrices  $C_n$ . Such procedures are called second-order because they usually leverage information from the Fisher information matrix (or the Hessian of the log-likelihood), also known as curvature information. They are also known as *adaptive* procedures because they adapt their hyperparameters, i.e., learning rates  $\gamma_n$  or condition matrices  $C_n$ , according to observed data. For instance, let  $C_n \equiv \mathcal{I}(\theta_\star)^{-1}$  and  $\gamma_1 = 1$ . Plugging in  $C_n = \mathcal{I}(\theta_\star)^{-1}$  in Theorem 3.2, the asymptotic variance of the SGD estimators is

$$(1/n)\gamma_1^2(2\gamma_1\mathcal{I}(\theta_\star)^{-1}\mathcal{I}(\theta_\star) - I)^{-1}\mathcal{I}(\theta_\star)^{-1}\mathcal{I}(\theta_\star)\mathcal{I}(\theta_\star)^{-1} = (1/n)\mathcal{I}(\theta_\star)^{-1},$$

which is the theoretically optimal asymptotic variance of the MLE, i.e., the Cramér-Rao lower bound.

Therefore, to achieve asymptotic efficiency, second-order procedures need to estimate the Fisher information matrix. Because  $\theta_*$  is unknown one can simply use  $C_n = \mathcal{I}(\theta_n^{\text{im}})^{-1}$  (or  $C_n = \mathcal{I}(\theta_{n-1}^{\text{sgd}})^{-1}$ ) as an iterative estimate of  $\mathcal{I}(\theta_*)$ , and the same optimality result holds. This approach in second-order explicit SGD was first studied by [Sakrison \(1965\)](#), and later by [Nevelson and Khasminskii \(1973, Chapter 8, Theorem 5.4\)](#). It was later extended by [Fabian \(1978\)](#) and several other authors. Notably, [Amari \(1998\)](#) refers to the direction  $\mathcal{I}(\theta_{n-1}^{\text{sgd}})^{-1} \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{sgd}})$  as the “natural gradient” and uses information geometry arguments to prove statistical optimality.

An alternative way to implement second-order procedures is to use stochastic approximation to estimate  $\mathcal{I}(\theta_*)$ , in addition to the approximation procedure estimating  $\theta_*$ . For example, [Amari et al. \(2000\)](#) proposed the following second-order procedure,

$$\begin{aligned} C_n^{-1} &= (1 - a_n)C_{n-1}^{-1} + a_n \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{am}}) \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{am}})^\top \\ \theta_n^{\text{am}} &= \theta_{n-1}^{\text{am}} + \gamma_n C_n \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{am}}), \end{aligned} \quad (10)$$

where  $a_n = a_1/n$  is a learning rate sequence, separate from  $\gamma_n$ . By standard stochastic approximation,  $C_n^{-1}$  converges to  $\mathcal{I}(\theta_*)$ , and thus procedure (10) is asymptotically optimal. However, there are two important problems with procedure (10). First, it is computationally costly because of matrix inversions. A faster way is to apply quasi-Newton ideas. SGD-QN developed by [Bordes et al. \(2009\)](#) is such a procedure where the first expensive matrix computations are substituted by typical secant conditions. Second, the stochastic approximation of  $\mathcal{I}(\theta_*)$  is usually very noisy in high-dimensional problems and this affects the main approximation for  $\theta_*$ . Recently, more robust variants of SGD-QN have been proposed ([Byrd et al., 2014](#)).

Another notable adaptive procedure is AdaGrad ([Duchi et al., 2011a](#)), which is defined as

$$\begin{aligned} C_n^{-1} &= C_{n-1}^{-1} + \text{diag} \left( \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{ada}}) \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{ada}})^\top \right), \\ \theta_n^{\text{ada}} &= \theta_{n-1}^{\text{ada}} + \gamma_1 C_n^{1/2} \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{ada}}), \end{aligned} \quad (11)$$

where  $\text{diag}(\cdot)$  takes the diagonal matrix of its matrix argument, and the learning rate is set constant to  $\gamma_n \equiv \gamma_1$ . AdaGrad can be considered a second-order procedure because it tries to approximate the Fisher information matrix, however it only uses gradient information so technically it is first-order. Under appropriate conditions,  $C_n^{-1} \rightarrow \text{diag}(\mathcal{I}(\theta_*))$  and a simple modification in the proof of Theorem 3.2 can show that the asymptotic variance of the AdaGrad estimate is given by

$$\sqrt{n} \text{Var}(\theta_n^{\text{ada}}) \rightarrow \frac{\gamma_1}{2} \text{diag}(\mathcal{I}(\theta_*))^{-1/2}. \quad (12)$$



This result reveals an interesting trade-off achieved by AdaGrad and a subtle contrast to first-order SGD procedures. The asymptotic variance of AdaGrad is  $O(1/\sqrt{n})$ , which indicates significant loss of information. However, this rate is attained *regardless* of the specification of the learning rate parameter  $\gamma_1$ .<sup>5</sup> In contrast, as shown in Theorem 3.2, first-order SGD procedures require  $2\gamma_1\mathcal{I}(\theta_\star) - I \succ 0$  in order to achieve the  $O(1/n)$  rate, and the rate is significantly worse if this condition is not met. For instance, Nemirovski et al. (2009) given an example of misspecification of  $\gamma_1$  where the rate of first-order explicit SGD is  $O(n^{-\epsilon})$ , and  $\epsilon$  can be arbitrarily small. The variance result (12) is illustrated in numerical experiments of Section 5.1.1.

### 3.3 Optimality with averaging

As shown in Section 3.2.1, Theorem 3.2 implies that first-order SGD procedures can be statistically inefficient, especially in many dimensions. One surprisingly simple idea to achieve statistical efficiency is to combine larger learning rates with averaging of the iterates. In particular, we consider the procedure

$$\begin{aligned}\theta_n^{\text{im}} &= \theta_{n-1}^{\text{im}} + \gamma_n \nabla \log f(Y_n; X_n, \theta_n^{\text{im}}), \\ \overline{\theta}_n^{\text{im}} &= \frac{1}{n} \sum_{i=1}^n \theta_i^{\text{im}},\end{aligned}\tag{13}$$

where  $\theta_n^{\text{im}}$  are the typical implicit SGD iterates (4), and  $\gamma_n = \gamma_1 n^{-\gamma}$ ,  $\gamma \in [0.5, 1)$ . Under suitable conditions, the iterates  $\overline{\theta}_n^{\text{im}}$  are asymptotically efficient. This is formalized in the following theorem.

**Theorem 3.3.** *Consider SGD procedure (13) and suppose Assumptions 3.1(a),(c),(d), and (e) hold, where  $\gamma \in [0.5, 1)$ . Then, the iterate  $\overline{\theta}_n^{\text{im}}$  converges to  $\theta_\star$  in probability and is asymptotically efficient, i.e.,*

$$n \text{Var} \left( \overline{\theta}_n^{\text{im}} \right) \rightarrow \mathcal{I}(\theta_\star)^{-1}.$$

*Remarks.* In the context of explicit stochastic approximations, averaging was first proposed and analyzed by Ruppert (1988) and Bather (1989). Ruppert (1988) argued that larger learning rates in stochastic approximation uncorrelates

---

<sup>5</sup> This follows from a technicality in Lemma D.1. On a high-level, the term  $\gamma_{n-1}/\gamma_n$  is important for the variance rates of AdaGrad and SGD. When  $\gamma_n \propto 1/n$ , as in Theorem 3.2, it holds  $\gamma_{n-1}/\gamma_n = 1 + \gamma_n/\gamma_1 + O(\gamma_n^2)$ , which explains the quantity  $2\mathcal{I}(\theta_\star) - I/\gamma_1$  in first-order SGD. The rate  $O(1/n)$  is attained only if  $2\mathcal{I}(\theta_\star) - I/\gamma_1 \succ 0$ . When  $\gamma_n \propto 1/\sqrt{n}$ , as in AdaGrad, it holds  $\gamma_{n-1}/\gamma_n = 1 + o(\gamma_n)$  and the rate  $O(1/\sqrt{n})$  is attained without any additional requirement.



the iterates allowing averaging to improve efficiency. Polyak and Juditsky (1992a) expanded the scope of averaging by proving asymptotic optimality in more general explicit stochastic approximations that operate under suitable conditions similar to Theorem 3.3. Polyak and Juditsky (1992a) thus proved that slowly-converging stochastic approximations can be improved by using larger learning rates and averaging of the iterates. Recent work has analyzed explicit updates with averaging (Zhang, 2004; Xu, 2011; Bach and Moulines, 2013; Shamir and Zhang, 2012), and has shown their superiority in numerous learning tasks.

### 3.4 Asymptotic normality

Asymptotic distributions, or more generally invariance principles, are well-studied in classical stochastic approximation (Ljung et al., 1992, Chapter II.8). In this section we leverage Fabian’s theorem (Fabian, 1968b) to show that iterates from implicit SGD are asymptotically normal.

**Theorem 3.4.** *Suppose that Assumptions 3.1(a),(c),(d),(e),(f) hold. Then, the iterate  $\theta_n^{\text{im}}$  of implicit SGD (4) is asymptotically normal, such that*

$$n^{\gamma/2}(\theta_n^{\text{im}} - \theta_\star) \rightarrow \mathcal{N}_p(0, \Sigma),$$

where  $\Sigma = \gamma_1^2 (2\gamma_1 C\mathcal{I}(\theta_\star) - I)^{-1} C\mathcal{I}(\theta_\star)C$ .

*Remarks.* The combined results of Theorems 3.1, 3.2, and 3.4 indicate that implicit SGD is numerically stable and has known asymptotic variance and distribution. Therefore, contrary to explicit SGD that has severe stability issues, implicit SGD emerges as a stable estimation procedure with known standard errors, which enables typical statistical tasks, such as confidence intervals, hypothesis testing, and model checking. We show empirical evidence supporting this claim in Section 5.1.2.

### 3.5 Stability

To illustrate the stability, or lack thereof, of both SGD estimators in small-to-moderate samples, we simplify the SGD procedures and inspect the size of the biases  $\mathbb{E}(\theta_n^{\text{sgd}} - \theta_\star)$  and  $\mathbb{E}(\theta_n^{\text{im}} - \theta_\star)$ . In particular, based on Theorem 3.1, we simply assume the Taylor expansion  $\nabla \log f(Y_n; X_n, \theta_n) = -\mathcal{I}(\theta_\star)(\theta_n - \theta_\star) + O(\gamma_n)$ ; to simplify further we ignore the remainder term  $O(\gamma_n)$ .

Under this simplification, the SGD procedures (3) and (4) can be written as follows:

$$\mathbb{E}(\theta_n^{\text{sgd}} - \theta_\star) = (I - \gamma_n \mathcal{I}(\theta_\star)) \mathbb{E}(\theta_{n-1}^{\text{sgd}} - \theta_\star) = P_1^n b_0, \quad (14)$$

$$\mathbb{E}(\theta_n^{\text{im}} - \theta_\star) = (I + \gamma_n \mathcal{I}(\theta_\star))^{-1} \mathbb{E}(\theta_{n-1}^{\text{im}} - \theta_\star) = Q_1^n b_0, \quad (15)$$

where  $P_1^n = \prod_{i=1}^n (I - \gamma_i \mathcal{I}(\theta_*))$ ,  $Q_1^n = \prod_{i=1}^n (I + \gamma_i \mathcal{I}(\theta_*))^{-1}$ , and  $b_0$  denotes the initial bias of the two procedures from a common starting point  $\theta_0$ . Thus, the matrices  $P_1^n$  and  $Q_1^n$  describe how fast the initial bias decays for the explicit and implicit SGD respectively. In the limit,  $P_1^n \rightarrow 0$  and  $Q_1^n \rightarrow 0$  (see proof of Lemma D.1), and thus both methods are *asymptotically stable*.

However, the explicit procedure has significant stability issues in small-to-moderate samples. By inspection of Eq. (14), the magnitude of  $P_1^n$  is dominated by  $\bar{\lambda}_f$ , the maximum eigenvalue of  $\mathcal{I}(\theta_*)$ . Furthermore, the rate of convergence is dominated by  $\lambda_f$ , the minimum eigenvalue of  $\mathcal{I}(\theta_*)$ .<sup>6</sup> For stability, it is desirable  $|1 - \gamma_1 \lambda_i| < 1$ , and for fast convergence  $|1 - \gamma_1 \lambda_i| \approx 0$ , for all eigenvalues  $\lambda_i \in \text{eig}(\mathcal{I}(\theta_*))$ . This roughly implies the requirements  $\gamma_1 < 2/\bar{\lambda}_f$  for stability, and  $\gamma_1 > 1/\lambda_f$  for convergence. This is problematic in high-dimensional settings because  $\bar{\lambda}_f$  is typically orders of magnitude larger than  $\lambda_f$ . Thus, the requirements for stability and speed of convergence are in conflict: to ensure stability we need a small learning rate parameter  $\gamma_1$ , thus paying a high price in convergence which will be at the order of  $O(n^{-\gamma_1 \lambda_f})$ , and vice versa.

In contrast, the implicit procedure is *unconditionally stable*. The eigenvalues of  $Q_1^n$  are  $\lambda'_i = \prod_{j=1}^n 1/(1 + \gamma_1 \lambda_i/j) = O(n^{-\gamma_1 \lambda_i})$ . Critically, it is no longer required to have a small  $\gamma_1$  for stability because the eigenvalues of  $Q_1^n$  are always less than one. We summarize these findings in the following lemma.

**Lemma 3.1.** *Let  $\bar{\lambda}_f = \max \text{eig}(\mathcal{I}(\theta_*))$ , and suppose  $\gamma_n = \gamma_1/n$  and  $\gamma_1 \bar{\lambda}_f > 1$ . Then, the maximum eigenvalue of  $P_1^n$  satisfies*

$$\max_{n>0} \max \{\text{eig}(P_1^n)\} = \Theta(2^{\gamma_1 \bar{\lambda}_f} / \sqrt{\gamma_1 \bar{\lambda}_f}).$$

*For the implicit method,*

$$\max_{n>0} \max \{\text{eig}(Q_1^n)\} = O(1).$$

*Remark.* Lemma 3.1 shows that in the explicit SGD procedure the effect from the initial bias can be amplified in an arbitrarily large way before fading out if the learning rate is misspecified (i.e., if  $\gamma_1 \gg 1/\bar{\lambda}_f$ ). This sensitivity of explicit SGD is well-known and requires problem-specific considerations to be avoided in practice e.g., pre-processing, small-sample tests, projections, truncation (Chen et al., 1987). In fact, there exists voluminous work, continuing up-to-date, in designing learning rates to stabilize explicit SGD; see, for example, a review by George and Powell (2006). Implicit procedures render such ad-hoc designs obsolete because they remain stable regardless of learning rate design, and still maintain the asymptotic convergence and efficiency properties of explicit SGD.

---

<sup>6</sup>To see this, note that the eigenvalues of  $P_1^n$  are  $\lambda'_i = \prod_{j=1}^n (1 - \gamma_1 \lambda_i/j) = O(n^{-\gamma_1 \lambda_i})$  if  $0 < \gamma_1 \lambda_i < 1$ . See also proof of Lemma 3.1.

## 4 Applications

In this section we show how to apply implicit SGD (4) for estimation in generalized linear models, Cox proportional hazards, and more general M-estimation problems. We start by developing an algorithm that efficiently computes the implicit updates (4), which is generally applicable to all aforementioned applications.

### 4.1 Efficient computation of implicit updates

The main difficulty in applying implicit SGD is the solution of the multidimensional fixed point equation (4). In a large family of models where the likelihood depends on the parameter  $\theta_*$  only through the natural parameter  $X_n^\top \theta_*$ , the solution of the fixed-point equation is feasible and computationally efficient. We prove the general result in Theorem 4.1, which depends on Assumption 3.1(b).

For the rest of this section we will treat  $\ell(X^\top \theta; Y)$  as a function of the natural parameter  $X^\top \theta$  (or  $X_n^\top \theta$  referring to the  $n$ th data point) for a fixed outcome  $Y$ . Thus,  $\ell'(X^\top \theta; Y)$  will refer to the first derivative of  $\ell$  with respect to  $X^\top \theta$  with fixed  $Y$ .

**Theorem 4.1.** *Suppose Assumption 3.1(b) holds. Then, the gradient for the implicit update (4) is a scaled version of the gradient at the previous iterate, i.e.,*

$$\nabla \log f(Y_n; X_n, \theta_n^{\text{im}}) = \lambda_n \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{im}}), \quad (16)$$

where the scalar  $\lambda_n$  satisfies,

$$\lambda_n \ell'(X_n^\top \theta_{n-1}^{\text{im}}; Y_n) = \ell'(X_n^\top \theta_{n-1}^{\text{im}} + \gamma_n \lambda_n \ell'(X_n^\top \theta_{n-1}^{\text{im}}; Y_n) X_n^\top C_n X_n; Y_n). \quad (17)$$

*Remarks.* Theorem 4.1 shows that the gradient  $\nabla \log f(Y_n; X_n, \theta_n^{\text{im}})$  in the implicit update (4) is a scaled gradient  $\nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{im}})$  calculated at the previous iterate  $\theta_{n-1}^{\text{im}}$ . Therefore, computing the implicit update reduces to finding the scale factor  $\lambda_n$  in Theorem 4.1. Under Assumption 3.1(b) narrow search bounds for  $\lambda_n$  are available. This leads to Algorithm 1 that is a generic implementation of implicit SGD for models satisfying Assumption 3.1(b). This implementation is fast because the interval  $B_n$  of search bounds in Algorithm 1 has size  $O(\gamma_n)$ .

### 4.2 Generalized linear models

In this section, we apply implicit SGD to estimate generalized linear models (GLMs). In such models,  $Y_n$  follows an exponential distribution conditional on

---

**Algorithm 1:** Efficient implementation of implicit SGD (4)

---

```

1: for all  $n \in \{1, 2, \dots\}$  do
2:   # compute search bounds  $B_n$ 
3:    $r_n \leftarrow \gamma_n \ell' (X_n^\top \theta_{n-1}^{\text{im}}; Y_n)$ 
4:    $B_n \leftarrow [0, r_n]$ 
5:   if  $r_n \leq 0$  then
6:      $B_n \leftarrow [r_n, 0]$ 
7:   end if
8:   # solve fixed-point equation by a root-finding method
9:    $\xi = \gamma_n \ell' (X_n^\top \theta_{n-1}^{\text{im}} + \xi X_n^\top C_n X_n; Y_n), \xi \in B_n$ 
10:   $\lambda_n \leftarrow \xi / r_n$ 
11:  # following update is equivalent to update (4)
12:   $\theta_n^{\text{im}} \leftarrow \theta_{n-1}^{\text{im}} + \gamma_n \lambda_n \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{im}})$ 
13: end for

```

---

$X_n$ , and  $\mathbb{E}(Y_n | X_n) = h(X_n^\top \theta_\star)$ , where  $h$  is the *transfer function* of the GLM model (Nelder and Wedderburn, 1972; Dobson and Barnett, 2008). Furthermore, the gradient of the GLM log-likelihood for parameter value  $\theta$  at data point  $(X_n, Y_n)$  is given by

$$\nabla \log f(Y_n; X_n, \theta) = [Y_n - h(X_n^\top \theta)] X_n. \quad (18)$$

The conditional variance of  $Y_n$  is  $\text{Var}(Y_n | X_n) = h'(X_n^\top \theta_\star) X_n X_n^\top$ , and thus the Fisher information matrix is  $\mathcal{I}(\theta) = \mathbb{E}(h'(X_n^\top \theta) X_n X_n^\top)$ .

Thus, SGD procedures (3) and (4) can be written as

$$\theta_n^{\text{sgd}} = \theta_{n-1}^{\text{sgd}} + \gamma_n C_n [Y_n - h(X_n^\top \theta_{n-1}^{\text{sgd}})] X_n, \quad (19)$$

$$\theta_n^{\text{im}} = \theta_{n-1}^{\text{im}} + \gamma_n C_n [Y_n - h(X_n^\top \theta_n^{\text{im}})] X_n. \quad (20)$$

Implementation of explicit SGD is straightforward. Implicit SGD can be implemented through Algorithm 1 because the conditions of Assumption 3.1(b) are fulfilled. In particular,  $\log f(Y; X, \theta) \equiv \ell(X^\top \theta; Y)$  where  $\ell(\eta; Y) = Y - h(\eta)$ . In typical models  $h$  is twice-differentiable and also  $h'(\eta) \geq 0$  because it is proportional to the conditional variance of  $Y$  given  $X$ , thus fulfilling Assumption 3.1(b). In the simplified case where  $C_n = I$ , the identity matrix, for all  $n$ , Algorithm 1 simplifies to Algorithm 2, which was first derived by Toulis et al. (2014). We make extensive experiments using Algorithm 2 in Section 5.2.

### 4.3 Cox proportional hazards

In this section, we apply SGD to estimate a Cox proportional hazards model, which is a popular model in survival analysis of censored failure times (Cox,

---

**Algorithm 2:** Estimation of GLMs with implicit SGD

---

```

1: for all  $n \in \{1, 2, \dots\}$  do
2:    $r_n \leftarrow \gamma_n [Y_n - h(X_n^\top \theta_{n-1}^{\text{im}})]$ 
3:    $B_n \leftarrow [0, r_n]$ 
4:   if  $r_n \leq 0$  then
5:      $B_n \leftarrow [r_n, 0]$ 
6:   end if
7:    $\xi = \gamma_n [Y_n - h(X_n^\top \theta_{n-1}^{\text{im}} + \xi \|X_n\|^2)], \xi \in B_n$ 
8:    $\theta_n^{\text{im}} \leftarrow \theta_{n-1}^{\text{im}} + \xi X_n$ 
9: end for

```

---

1972; Klein and Moeschberger, 2003). Multiple variations of the model exist, but we will analyze one simple variation that is popular in practice (Davison, 2003). Consider  $N$  individuals, indexed by  $i$ , with observed survival times  $Y_i$ , failure indicators  $d_i$ , and covariates  $X_i$ . The survival times can be assumed ordered,  $Y_1 < Y_2 \dots < Y_N$ , whereas  $d_i = 1$  denotes failure (e.g., death) and  $d_i = 0$  indicates censoring (e.g., patient dropped out of study). Given a failure for unit  $i$  ( $d_i = 1$ ) at time  $Y_i$ , the *risk set*  $\mathcal{R}_i$  is defined as the set of individuals that could possibly fail at  $Y_i$ , i.e., all individuals except those who failed or were censored before  $Y_i$ . In our simplified model,  $\mathcal{R}_i = \{i, i+1, \dots, N\}$ . Define  $\eta_i(\theta) \stackrel{\text{def}}{=} \exp(X_i^\top \theta)$ , then the log-likelihood  $\ell$  for  $\theta$  is given by (Davison, 2003, Chapter 10)

$$\ell(\theta; X, Y) = \sum_{i=1}^N [d_i - H_i(\theta) \eta_i(\theta)] X_i, \quad (21)$$

where  $H_i(\theta) = \sum_{j:i \in \mathcal{R}_j} d_j (\sum_{k \in \mathcal{R}_j} \eta_k(\theta))^{-1}$ . In an online setting, where  $N$  is infinite and data points  $(X_i, Y_i)$  are observed one at a time, future observations affect the likelihood of previous ones, as can be seen by inspection of Eq. (21). Therefore, we apply SGD assuming fixed  $N$  to estimate the MLE  $\theta_N^{\text{mle}}$ . As mentioned in Section 1, our theory in Section 3 can be applied unchanged if we only substitute  $\theta_*$ , the true parameter, with the MLE  $\theta_N^{\text{mle}}$ .

A straightforward implementation of explicit SGD (3) for the Cox model is shown in Algorithm 3. For implicit SGD (4) we have the update

$$\theta_n^{\text{im}} = \theta_{n-1}^{\text{im}} + \gamma_n [d_i - H_i(\theta_n^{\text{im}}) \eta_i(\theta_n^{\text{im}})] X_i, \quad (22)$$

which is similar to the implicit procedure for GLMs (20). However, the log-likelihood term  $d_i - H_i(\theta_n^{\text{im}}) \eta_i(\theta_n^{\text{im}})$  does not satisfy the conditions of Assumption 3.1(b). Although the term  $\eta_i(\theta)$  is increasing with respect to  $X_i^\top \theta$ ,  $H_i(\theta)$  may be increasing or may be decreasing because it depends on terms  $X_j^\top \theta, j \neq i$ , as

well. Thus, Theorem 4.1 cannot be applied. One way to circumvent this problem and apply Theorem 4.1 is to simply compute  $H_i(\cdot)$  on the previous update  $\theta_{n-1}^{\text{im}}$  instead of the current  $\theta_n^{\text{im}}$ . Then, update (22) becomes,

$$\theta_n^{\text{im}} = \theta_{n-1}^{\text{im}} + \gamma_n [d_i - H_i(\theta_{n-1}^{\text{im}}) \eta_i(\theta_n^{\text{im}})] X_i, \quad (23)$$

which now satisfies Assumption 3.1(b) as  $H_i(\theta_{n-1}^{\text{im}})$  is constant with respect to  $\theta_n^{\text{im}}$ .<sup>7</sup> The implicit SGD procedure for fitting Cox models is shown in Algorithm 4. We run experiments with implicit SGD on Cox models with simulated data in Section 5.2.

Algorithm 3: Explicit SGD for Cox model	Algorithm 4: Implicit SGD for Cox model
<b>1</b> for $n = 1, 2, \dots$ do <b>2</b> $i \leftarrow \text{sample}(1, N)$ <b>3</b> $\hat{H}_i \leftarrow \sum_{j:i \in \mathcal{R}_j} \frac{d_j}{\sum_{k \in \mathcal{R}_j} \eta_k(\theta_{n-1}^{\text{sgd}})}$ <b>4</b> $w_{n-1} \leftarrow [d_i - \hat{H}_i \eta_i(\theta_{n-1}^{\text{sgd}})]$ <b>5</b> $\theta_n^{\text{sgd}} = \theta_{n-1}^{\text{sgd}} + \gamma_n w_{n-1} C_n X_i$	<b>1</b> for $n = 1, 2, \dots$ do <b>2</b> $i \leftarrow \text{sample}(1, N)$ <b>3</b> $\hat{H}_i \leftarrow \sum_{j:i \in \mathcal{R}_j} \frac{d_j}{\sum_{k \in \mathcal{R}_j} \eta_k(\theta_{n-1}^{\text{im}})}$ <b>4</b> $w(\theta) \stackrel{\text{def}}{=} d_i - \hat{H}_i \eta_i(\theta)$ <b>5</b> $W_n \leftarrow w(\theta_{n-1}^{\text{im}}) C_n X_i$ <b>6</b> $\lambda_n w(\theta_{n-1}^{\text{im}}) = w(\theta_{n-1}^{\text{im}} + \gamma_n \lambda_n W_n)$ <b>7</b> $\theta_n^{\text{im}} = \theta_{n-1}^{\text{im}} + \gamma_n \lambda_n W_n$

## 4.4 M-Estimation

Given  $N$  observed data points  $(X_i, Y_i)$  and a convex function  $\rho : \mathbf{R} \rightarrow \mathbf{R}^+$ , the *M-estimator* is defined as

$$\hat{\theta}^m = \arg \min_{\theta} \sum_{i=1}^N \rho(Y_i - X_i^\top \theta), \quad (24)$$

where it is assumed  $Y_i = X_i^\top \theta_* + \epsilon_i$ , and  $\epsilon_i$  are i.i.d. zero mean-valued noise. M-estimators are especially useful in robust statistics (Huber et al., 1964; Huber, 2011) because appropriate choice of  $\rho$  can reduce the influence of outliers in data. Typically,  $\rho$  is twice-differentiable around zero. In this case,

$$\mathbb{E} \left( \rho'(Y_i - X_i^\top \hat{\theta}^m) X_i \right) = 0, \quad (25)$$

where the expectation is over the empirical data distribution. Thus, according to Section 1, SGD procedures can be applied to approximate the M-estimator

<sup>7</sup>This idea can be more generally used in order to apply implicit SGD on models that do not satisfy Assumption 3.1(b); see Section 6 for a discussion.

$\hat{\theta}^m$ .<sup>8</sup> There has been increased interest in the literature for fast approximation of M-estimators due to the robustness properties of such estimators (Donoho and Montanari, 2013; Jain et al., 2014).

The implicit SGD procedure for approximating M-estimators is defined in Algorithm 5, and is a simple adaptation of Algorithm 1.

---

**Algorithm 5:** Implicit SGD for M-estimation

---

```

1 for  $n = 1, 2, \dots$  do
2    $i \leftarrow \text{sample}(1, N)$ 
3    $w(\theta) \stackrel{\text{def}}{=} \rho'(Y_i - X_i^\top \theta)$ 
4    $\lambda_n w(\theta_{n-1}^{\text{im}}) \leftarrow w(\theta_{n-1}^{\text{im}} + \gamma_n \lambda_n w(\theta_{n-1}^{\text{im}}) C_n X_i)$    # implicit update
5    $\theta_n^{\text{im}} \leftarrow \theta_{n-1}^{\text{im}} + \gamma_n \lambda_n w(\theta_{n-1}^{\text{im}}) C_n X_i$ 

```

---

Importantly,  $\rho$  is convex and thus  $\rho'' \geq 0$  and therefore the conditions of Assumption 3.1(b) are met. Thus, Step 4 of Algorithm 5 is a straightforward application of Algorithm 1 by simply setting  $\ell'(X_n' \theta_{n-1}^{\text{im}}; Y_n) \equiv \rho'(Y_n - X_n^\top \theta_n^{\text{im}})$ . The asymptotic variance of  $\theta_n^{\text{im}}$  is also easy to derive. If  $S \stackrel{\text{def}}{=} \mathbb{E}(X_n X_n^\top)$ ,  $C_n \rightarrow C > 0$  such that  $S$  and  $C$  commute,  $\psi^2 \stackrel{\text{def}}{=} \mathbb{E}(\rho'(\epsilon_i)^2)$ , and  $v(z) \stackrel{\text{def}}{=} \mathbb{E}(\rho'(\epsilon_i + z))$ , Theorem 3.2 can be leveraged to show that

$$n \text{Var}(\theta_n^{\text{im}}) \rightarrow \psi^2 (2v'(0)CS - I)^{-1} CSC. \quad (26)$$

Historically, one of the first applications of explicit stochastic approximation procedures in robust estimation was due to Martin and Masreliez (1975). The asymptotic variance (26) was first derived, only for the explicit SGD case, by Poljak and Tsytkin (1980) using stochastic approximation theory from Nevelson and Khasminskii (1973).

## 5 Simulation and data analysis

In this section, we demonstrate the computational and statistical advantages of SGD estimation procedures (3) and (4). For our experiments we developed a new R package, namely `sgd`, which has been published on CRAN.<sup>9</sup>

---

<sup>8</sup> It is also typical to assume that the density of  $\epsilon_i$  is symmetric around zero. Therefore, it also holds  $\mathbb{E}(\rho'(Y_i - X_i^\top \theta_\star) X_i) = 0$ , where the expectation is over the true data distribution. According to Section 1 SGD procedures can also be used to estimate  $\theta_\star$  in the case of infinite stream of observations ( $N = \infty$ ). In this section we only consider the case of finite  $N$ , but it is trivial to adapt our procedures to infinite  $N$ .

<sup>9</sup>Our R package resides on CRAN at <http://cran.r-project.org/web/packages/sgd/index.html>, and has been co-authored with Dustin Tran and Kuang Ye. All experiments

The experiments are split into three sets. In the first set, presented in Section 5.1, we aim to validate the theoretical results of Section 3. In the second set, presented in Section 5.2, we aim to validate the performance of SGD procedures. We focus particularly on implicit SGD because it is a more stable estimation procedure, whereas for explicit SGD we could not devise a learning rate design that worked uniformly well in all experiments. The third set of experiments, presented in the Appendix, focuses on comparisons of implicit SGD, including implicit SGD with averaging, against popular machine learning methods, such as averaged explicit SGD on a large support vector machine (SVM) model, **prox-SAG** (Schmidt et al., 2013), **prox-SVRG** (Xiao and Zhang, 2014), and averaged explicit SGD (Xu, 2011; Bach and Moulines, 2013), on typical machine learning tasks.

## 5.1 Validation of theory

In this section we aim to validate the theoretical results of Section 3, namely the result on asymptotic variance (Theorem 3.2 and asymptotic normality (Theorem 3.4) of SGD procedures.

### 5.1.1 Asymptotic variance

In this experiment we use a normal linear model following the experimental setup of Xu (2011) to check the asymptotic variance results of Theorem 3.2. The procedures we test are explicit SGD (3), implicit SGD (4), and AdaGrad (11). For simplicity we use first-order SGD where  $C_n \equiv I$ .

In the experiment we calculate the empirical variance of said procedures for 25 values of their common learning rate parameter  $\gamma_1$  in the interval  $[1.2, 10]$ . For every value of  $\gamma_1$  we calculate the empirical variances through the following process, repeated for 150 times. First, we set  $\theta_\star = (1, 1, \dots, 1)^\top \in \mathbf{R}^{20}$  as the true parameter value. For iterations  $n = 1, 2, \dots, 1500$ , we sample covariates as  $X_n \sim \mathcal{N}_p(0, S)$ , where  $S$  is diagonal with elements uniformly in  $[0.5, 5]$ . The outcome  $Y_n$  is then sampled as  $Y_n|X_n \sim \mathcal{N}(X_n^\top \theta_\star, 1)$ . In every repetition we store the iterate  $\theta_{1500}$  for every tested procedure and then calculate the empirical variance of stored iterates over all 150 repetitions.

For any fixed learning rate parameter  $\gamma_1$ , we set the learning rates as follows. For implicit SGD we set  $\gamma_n = \gamma_1/n$ , and for AdaGrad we set  $\gamma_n = \gamma_1$ , as it is typical. However, for explicit SGD we set  $\gamma_n = \min(0.3, \gamma_1/(n + \|X_n\|^2))$  in order to stabilize its updates. This trick is necessary by the analysis of Section 3.5. In particular, the Fisher information matrix is  $\mathcal{I}(\theta_\star) = \mathbb{E}(X_n X_n^\top) = S$ , and thus the minimum eigenvalue is  $\underline{\lambda}_f = 0.5$  and the maximum is  $\bar{\lambda}_f = 5$ . Therefore,

---

were conducted on a single laptop running Linux Ubuntu 13.x with 8 cores@2.4GHz, 16Gb of RAM memory and 256Gb of physical storage with SSD technology.



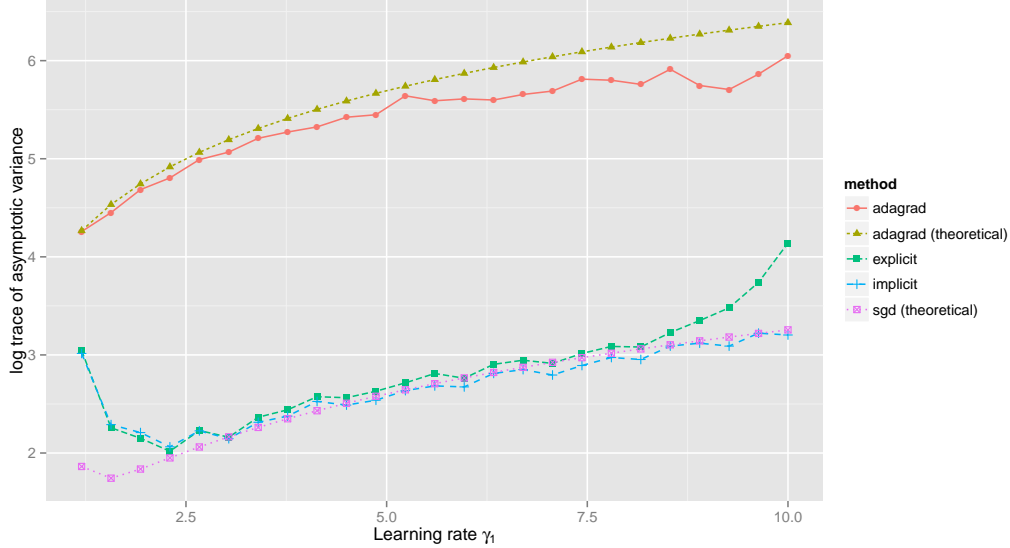


Figure 1: Simulation with normal model. The x-axis corresponds to learning rate parameter  $\gamma_1$ ; the y-axis curves corresponds to log trace of the empirical variance of tested procedures (explicit/implicit SGD, AdaGrad). Theoretical asymptotic variances of SGD and AdaGrad are plotted as well. Implicit SGD is stable and its empirical variance is very close to its asymptotic value. Explicit SGD becomes unstable at large  $\gamma_1$ . AdaGrad is statistically inefficient but remains stable to large learning rates.

for stability we require  $\gamma_1 < 1/\bar{\lambda}_f = 0.2$  and for fast convergence we require  $\gamma_1 < 1/(2\lambda_f) = 1$ . The two requirements are incompatible, which indicates that explicit SGD can have serious stability issues.

For given  $\gamma_1 > 1$ , the asymptotic variance of SGD procedures after  $n$  iterations is  $(1/n)\gamma_1^2(2\gamma_1 S - I)^{-1}S$ , by Theorem 3.2. The asymptotic variance of AdaGrad after  $n$  iterations is equal to  $(\gamma_1/2\sqrt{n})S^{-1/2}$ , by the analysis of Section 3.2.1. The log traces of the empirical variance of the SGD procedures and AdaGrad in this experiment are shown in Figure 9. The x-axis corresponds to different values of the learning rate parameter  $\gamma_1$ , and the y-axis corresponds to the log trace of the empirical variance of the iterates for all three different procedures. We also include curves for the aforementioned theoretical values of the empirical variances.

We see that our theory predicts well the empirical variance of all methods. Explicit SGD performs on par with implicit SGD for moderate values of  $\gamma_1$ , however, it required a modification in its learning rate to make it work. Furthermore, explicit SGD quickly becomes unstable at larger values of  $\gamma_1$  (see, for example, its

empirical variance for  $\gamma_1 = 10$ ), and in several instances, not considered in Figure 9, it numerically diverged. On the other hand, AdaGrad is stable to the specification of  $\gamma_1$  and tracks its theoretical variance well. However, it gives inefficient estimators because their variance has order  $O(1/\sqrt{n})$ . Implicit SGD effectively combines stability and good statistical efficiency. First, it remains very stable to the entire range of the learning rate parameter  $\gamma_1$ . Second, its empirical variance is  $O(1/n)$  and it tracks closely the theoretical value predicted by Theorem 3.2 for all  $\gamma_1$ .

### 5.1.2 Asymptotic normality

In this experiment we use the normal linear model in the setup of Section 5.1.1 to check the asymptotic normality result of Theorem 3.4. We only test first-order implicit SGD (4) and first-order explicit SGD for simplicity. Asymptotic normality for explicit stochastic approximations has been shown Fabian (1968a) and several other authors (see also Ljung et al., 1992, Parts I, II).

In the experiment we define a set learning rates  $(0.5, 1, 3, 5, 6, 7)$ . For every learning rate and for every method, implicit or explicit SGD, we take 400 samples of  $N(\theta_N^{\text{im}} - \theta_\star)^\top \Sigma^{-1}(\theta_N^{\text{im}} - \theta_\star)$ , where  $N = 1200$ ; i.e., we run each SGD procedure for 1200 iterations. The matrix  $\Sigma$  is the asymptotic variance matrix in Theorem 3.4, and  $\theta_\star \stackrel{\text{def}}{=} 10 \exp(-2 \cdot (1, 2, \dots, p))$ , is the true parameter value. We use the ground-truth values both for  $\Sigma$  and  $\theta_\star$ , as we are only interested to test normality of the iterates in this experiment. We also tried  $p = 5, 10, 100$  as the parameter dimension. Because the explicit SGD was very unstable across experiments we only report results for  $p = 5$ . Results on the implicit procedure for larger  $p$  are given in Appendix G.1, where we also include results for a logistic regression model.

By Theorem 3.4 the quantity  $N(\theta_N^{\text{im}} - \theta_\star)^\top \Sigma^{-1}(\theta_N^{\text{im}} - \theta_\star)$  is a chi-squared random variable with  $p$  degrees of freedom. Thus, for every procedure we plot this quantity against independent samples from a  $\chi_p^2$  distribution and visually check for deviations. As before, we tried to stabilize explicit SGD as much as possible by setting  $\gamma_n = \min(0.3, \gamma_1/(n + \|X_n\|^2))$  as the learning rate. This worked in many iterations, but not for all. Iterations for which explicit SGD diverged were not considered. For implicit SGD we simply set  $\gamma_n = \gamma_1/n$  without additional tuning.

The results of this experiment are shown in Figure 2. The vertical axis on the grid corresponds to different values of the learning rate parameter  $\gamma_1$ , and the horizontal axis has histograms of  $N(\theta_N - \theta_\star)^\top \Sigma^{-1}(\theta_N - \theta_\star)$  for both implicit and explicit procedures, and also includes samples from a  $\chi_5^2$  distribution for visual comparison.

We see that the distribution  $N(\theta_N^{\text{im}} - \theta_\star)^\top \Sigma^{-1}(\theta_N^{\text{im}} - \theta_\star)$  of the implicit iter-

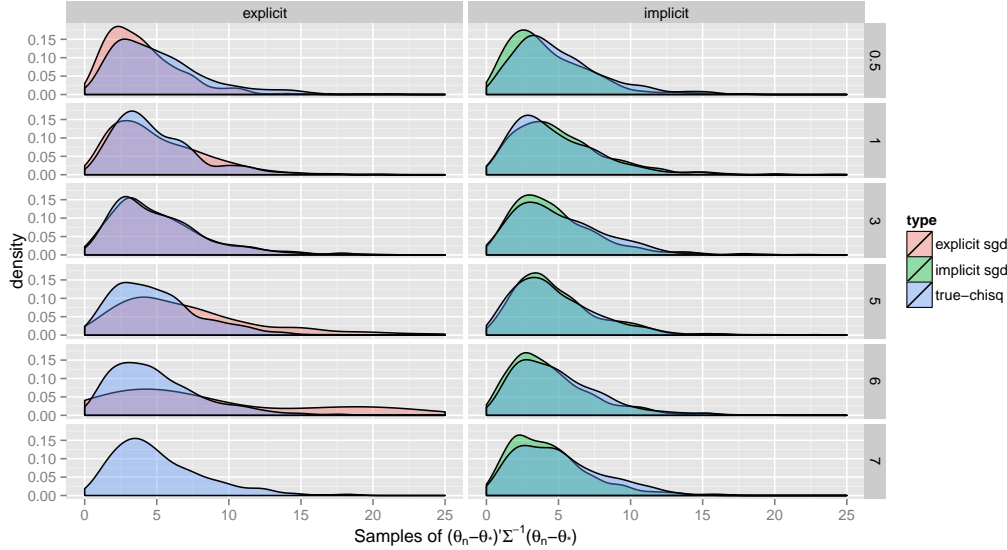


Figure 2: Simulation with normal model. The x-axis corresponds to the SGD procedure (explicit or implicit) for various values of the learning rate parameter,  $\gamma_1 \in \{0.5, 1, 3, 5, 7\}$ . The histograms (x-axis) for the SGD procedures are 500 replications of SGD where at each replication we only store the quantity  $N(\theta_N - \theta_\star)^\top \Sigma^{-1}(\theta_N - \theta_\star)$ , for every method ( $N = 1200$ ); the theoretical covariance matrix  $\Sigma$  is different for every learning rate and is given in Theorem 3.2. The data generation model is the same as in Section 5.1.1. We observe that implicit SGD is stable and follows the nominal chi-squared distribution. Explicit SGD becomes unstable at larger  $\gamma_1$  and its distribution does not follow the nominal one well. In particular, the distribution of  $N(\theta_N^{\text{sgd}} - \theta_\star)^\top \Sigma^{-1}(\theta_N^{\text{sgd}} - \theta_\star)$  becomes increasingly heavy-tailed as the learning rate parameter gets larger, and eventually diverges for  $\gamma_1 \geq 7$ .

ates follows the nominal chi-squared distribution. This also seems to be unaffected by the size learning rate parameter. However, the distribution  $N(\theta_N^{\text{sgd}} - \theta_\star)^\top \Sigma^{-1}(\theta_N^{\text{sgd}} - \theta_\star)$  of the explicit iterates does not follow a chi-squared distribution, expect for small learning rate parameter values. For example, as the learning rate parameter increases, the distribution becomes more heavy-tailed (e.g., for  $\gamma_1 = 6$ ), indicating that explicit SGD becomes unstable. Particularly for  $\gamma_1 = 7$  explicit SGD diverged in almost all replications, and thus a histogram could not be constructed.

## 5.2 Validation of performance

In this section we aim to validate the performance of implicit SGD estimation against deterministic estimation procedures that are optimal. In Section 5.2.1, we compare implicit SGD with R’s `glm()` function, which calculates the theoretically optimal MLE. In this experiment, we wish to test the computational efficiency of implicit SGD in terms of the problem size, i.e., parameter dimension  $p$ , and number of data points  $N$ , and its statistical efficiency in terms of MSE. We perform similar tests in Section 5.2.2 but on larger problem sizes, where we compare with R’s `biglm` package, which is popular in estimating GLM models in large data sets (large  $N$ , small  $p$ ). In Section 5.2.3 we compare implicit SGD with `glmnet` of Friedman et al. (2010), which is an efficient implementation of the elastic net for several GLMs for problems with  $N > p$ . In Sections 5.2.6, we re-analyze data from the NMMAAPS study (Samet et al., 2000) and show how implicit SGD can be naturally extended to fit a large generalized additive model (GAM) (Hastie and Tibshirani, 1990) to estimate the effects of air pollution on public health. We also compare with recently published statistical methods that are specifically designed to fit large-scale GAMs, and demonstrate SGD’s superior performance.

In summary, our experimental results demonstrate that implicit SGD performs on par with optimal deterministic methods, even in problems with moderate-to-large data sets where stochastic methods are usually avoided. Combined its strong theoretical guarantees that were validated in Section 5.1, implicit SGD emerges as a principled estimation method that can be the workhorse of efficient estimation with large data sets in statistical practice.

### 5.2.1 Experiments with `glm()` function

The built-in function `glm()` in R<sup>10</sup> performs deterministic maximum-likelihood estimation through iterative reweighted least squares. In this experiment, we wish to compare computing time and MSE between first-order implicit SGD and `glm()`. Our simulated data set is a simple normal linear model constructed as follows. First, we sample a binary  $p \times p$  design matrix  $X = (x_{ij})$  such that  $x_{i1} = 1$  (intercept) and  $P(x_{ij} = 1) = s$  i.i.d, where  $s \in (0, 1)$  determines the sparsity of  $X$ . We set  $s = 0.08$  indicating that roughly 8% of the  $X$  matrix will be nonzero. We generate  $\theta_*$  by sampling  $p$  elements from  $(-1, -0.35, 0, 0.35, 1)$  with replacement. The outcomes are  $Y_i = X_i^\top \theta_* + \epsilon_i$ , where  $\epsilon_i \sim \mathcal{N}(0, 1)$  i.i.d.,

---

<sup>10</sup>Documentation is available at <http://stat.ethz.ch/R-manual/R-patched/library/stats/html/glm.html>.

Table 1: Parameters from regressing computation time and MSE against  $(N, p)$  in log-scale for `glm()` and implicit GLM. Computation time for `glm()` is roughly  $O(p^{1.47}N)$  and for implicit SGD, it is  $O(p^{0.2}N^{0.9})$ . Implicit SGD scales better in parameter dimension  $p$ , whereas MSE for both methods are comparable, at the order of  $O(\sqrt{p/N})$ .

METHOD	TIME(SEC)		MSE	
	log $p$ (SE)	log $N$ (SE)	log $p$ (SE)	log $N$ (SE)
<b>GLM()</b> FUNCTION	1.46 (0.019)	1.03 (0.02)	0.52 (0.007)	-0.52 (0.006)
IMPLICIT SGD	0.19 (0.012)	0.9 (0.01)	0.58 (0.007)	-0.53 (0.006)

and  $X_i = (x_{ij})$  is the  $p \times 1$  vector of  $i$ 's covariates. By GLM properties,

$$\mathcal{I}(\theta_*) = \mathbb{E} (h'(X_i^\top \theta_*) X_i X_i^\top) = \begin{pmatrix} 1 & s & s & \cdots & s \\ s & s & s^2 & \cdots & s^2 \\ s & s^2 & s & s^2 & \cdots \\ \cdots & s^2 & \cdots & s & \cdots \\ s & s^2 & \cdots & \cdots & s \end{pmatrix}.$$

Slightly tedious algebra can show that the eigenvalues of  $\mathcal{I}(\theta_*)$  are  $s(1-s)$  with multiplicity  $(p-2)$  and the two solutions of  $x^2 - A(s)x + B(s) = 0$ , where  $A(s) = 1 + s + s^2(p-2)$  and  $B(s) = s(1-s)$ . It is thus possible to use the analysis of Section 3.2 and Eq. (9) to derive a theoretically optimal learning rate. We sample 200 pairs  $(p, N)$  for the problem size, uniformly in the ranges  $p \sim [10, 500]$  and  $N \sim [500, 50000]$ , and obtain running times and MSE of the estimates from implicit SGD and `glm()`. Finally, we then run a regression of computing time and MSE against the problem size  $(N, p)$ .

The results are shown in Table 1. We observe that implicit SGD scales better in both sample size  $N$ , and especially in the model size  $p$ . We also observe, that this significant computational gain does not come with much efficiency loss. In fact, averaged over all samples, the MSE of the implicit SGD is on average 10% higher than the MSE of `glm()` function with a standard error of  $\pm 0.005$ . Furthermore, the memory requirements (not reported in Table 1) are roughly  $O(Np^2)$  for `glm()` and only  $O(p)$  for implicit SGD.

### 5.2.2 Experiments with `biglm`

The package `biglm` is a popular choice to fitting GLMs in large data sets (large  $N$ , small  $p$ ), and is part of the High-Performance Computing (HPC) task view

Table 2: Comparison of implicit SGD with **biglm**. MSE is defined as  $\|\theta_N - \theta_\star\|/\|\theta_0 - \theta_\star\|$ . Values “\*” indicate out-of-memory errors. **biglm** was run in combination with the **ffdf** package to map big data files to memory. Implicit SGD used a similar but slower ad-hoc method. The table reports computation times excluding file access.

$p$	$N$	SIZE (GB)	PROCEDURE			
			<b>BIGLM</b>		IMPLICIT SGD	
			TIME(SECS)	MSE	TIME(SECS)	MSE
1E2	1E5	0.021	2.32	0.028	2.4	0.028
1E2	5E5	0.103	8.32	0.012	7.1	0.012
1E2	1E6	0.206	16	0.008	14.7	0.009
1E2	1E7	2.1	232	0.002	127.9	0.002
1E2	1E8	20.6	*	*	1397	0.00
1E3	1E6	2.0	*	*	31.38	0.153
1E4	1E5	2.0	*	*	25.05	0.160

of the CRAN project.<sup>11</sup> It works in an iterative way by splitting the data set in many parts, and by updating the model parameters using incremental QR decomposition (Miller, 1992), which results in only  $O(p^2)$  memory requirement. In this experiment, we compare implicit SGD with **biglm** on larger data sets of Section 5.2.1. In particular, we focus on a few cases with small  $p$  and with large  $N$  such that  $N \cdot p$  remains roughly constant.

The results are shown in Table 2. We observe that implicit SGD is significantly faster at a very small efficiency loss. The difference is more dramatic at large  $p$ ; for example, when  $p = 10^3$  or  $p = 10^4$ , **biglm** quickly runs out of memory, whereas implicit SGD works without problems.

### 5.2.3 Experiments with glmnet

The **glmnet** package in R (Friedman et al., 2010) is a deterministic optimization algorithm for generalized linear models that uses the elastic net. It performs a component-wise update of the parameter vector, utilizing thresholding from the regularization penalties for more computationally efficient updates. One update

<sup>11</sup>See <http://cran.r-project.org/web/packages/biglm/index.html> for the **biglm** package. The HPC view of the CRAN project is here <http://cran.r-project.org/web/views/HighPerformanceComputing.html>.

over all parameters costs roughly  $O(Np)$  operations. Additional computational gains are achieved when the design matrix is sparse because fewer components are updated per each iteration.

In this experiment, we compare implicit SGD with **glmnet** on a subset of experiments in the original package release (Friedman et al., 2010). In particular, we implement the experiment of subsection 5.1 in that paper as follows. First, we sample the design matrix  $X \sim \mathcal{N}_p(0, \Sigma)$ , where  $\Sigma = b^2 U + I$  and  $U$  is the  $p \times p$  matrix of ones. The parameter  $b = \sqrt{\rho/(1-\rho)}$ , where  $\rho$  is the target correlation of columns of  $X$ , is controlled in the experiments. The outcomes are  $Y = X\theta_\star + \sigma^2\epsilon$ , where  $\theta_j^\star = (-1)^j \exp(-2(j-1)/20)$ , and  $\epsilon$  is a standard  $p$ -variate normal. The parameter  $\sigma$  is tuned to achieve a pre-defined signal-noise ratio. We report average computation times in Table 5 over 10 replications, which expands Table 1 of Friedman et al. (2010).

First, we observe that implicit SGD is consistently faster than the **glmnet** method. In particular, the SGD method scales better at larger  $p$  following a sublinear growth as noted in Section 5.2.1. Interestingly, it is also not affected by covariate correlation, whereas **glmnet** gets slower as more components need to be updated at every iteration. For example, with correlation  $\rho = 0.9$  and  $N = 1e5$ ,  $p = 200$ , the SGD method is almost 10x faster.

Second, to compare **glmnet** with implicit SGD in terms of MSE, we picked the median MSE produced by the grid of regularization parameters computed by **glmnet**. Because of regularization, it is reasonable to expect overall a slightly better performance for **glmnet** in situations where  $N$  is small compared to  $p$ . However, implicit SGD seems to perform better against the median performance of **glmnet**. This is reasonable because implicit SGD performs indirect regularization, as discussed in Section 6. Furthermore, Table 5 indicates a clear trend where, for bigger dimensions  $p$  and higher correlation  $\rho$ , implicit SGD is performing better than **glmnet** in terms of efficiency as well. We obtain similar results in a comparison on a logistic regression model, presented in Appendix G.3.

#### 5.2.4 Cox proportional hazards

In this experiment we test the performance of implicit SGD on estimating the parameters of a Cox proportional hazards model in a setup that is similar to the numerical example of Simon et al. (2011, Section 3).

We consider  $N = 1000$  units with covariates  $X \sim \mathcal{N}_N(0, \Sigma)$ , where  $\Sigma = 0.2U + I$ , and  $U$  is the matrix of ones. We sample times as  $Y_i \sim \text{Expo}(\eta_i(\theta_\star))$ , where  $\eta_i(\theta) = \exp(X_i^\top \theta)$ , and  $\theta_\star = (\theta_{\star,k})$  is a vector with  $p = 20$  elements defined as  $\theta_{\star,k} = 2(-1)^{-k} \exp(-0.1k)$ . Time  $Y_i$  is censored, and thus  $d_i = 0$ , according to probability  $(1 + \exp(-a(Y_i - b)))^{-1}$ , where  $b$  is a quantile of choice (set here as  $b = 0.8$ ), and  $a$  is set such that  $\min\{Y_i\}$  is censored with a prespecified

Table 3: Comparing implicit SGD with **glmnet**. Table reports running times (in secs.) and MSE for both procedures. The MSE of **glmnet** is calculated as the median MSE over the 100 grid values of regularization parameter computed by default (Friedman et al., 2010).

METHOD	METRIC	CORRELATION ( $\rho$ )			
		0	0.2	0.6	0.9
<hr/>					
$N = 1000, p = 10$					
<hr/>					
GLMNET	TIME(SEC)	0.005	0.005	0.008	0.022
	MSE	0.083	0.085	0.099	0.163
SGD	TIME(SEC)	0.011	0.011	0.011	0.011
	MSE	0.042	0.042	0.049	0.053
<hr/>					
$N = 5000, p = 50$					
<hr/>					
GLMNET		0.058	0.067	0.119	0.273
		0.044	0.046	0.057	0.09
SGD		0.059	0.056	0.057	0.057
		0.019	0.02	0.023	0.031
<hr/>					
$N = 100000, p = 200$					
<hr/>					
GLMNET		2.775	3.017	4.009	10.827
		0.017	0.017	0.021	0.033
SGD		1.475	1.464	1.474	1.446
		0.004	0.004	0.004	0.006

probability (set here as 0.1%). We replicate 50 times the following process. First, we run implicit SGD for  $2N$  iterations, and then measure MSE  $||\theta_n^{\text{im}} - \theta_*||^2$ , for all  $n = 1, 2, \dots, 2N$ . To set the learning rates we use Eq. (9), where the Fisher matrix is diagonally approximated, through the AdaGrad procedure (11). We then take the 5%, 50% and 95% quantiles of MSE across all repetitions and plot them against iteration number  $n$ .

The results are shown in Figure 5.2.4. In the figure we also plot (horizontal dashed lines) the 5% and 95% quantiles of the MSE of the MLE, assumed to be the best MSE achievable for SGD. We observe that Implicit SGD performs well compared to MLE, even in this small-sized problem. In particular, implicit SGD, under the aforementioned generic tuning of learning rates, converges to the



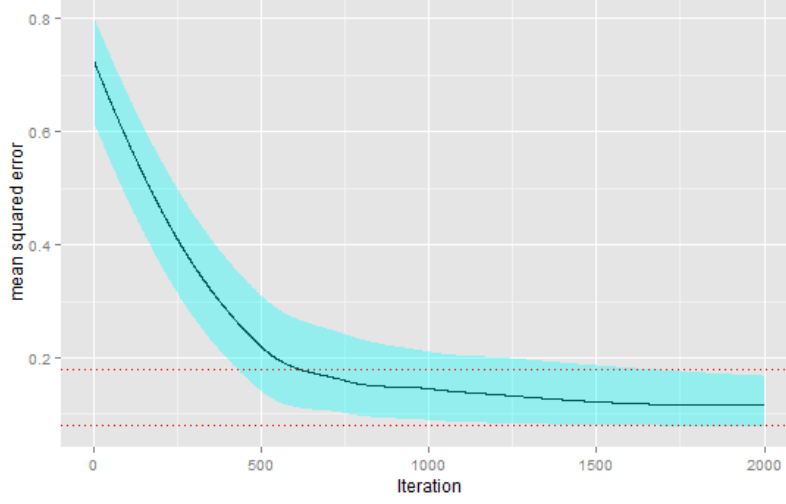


Figure 3: Implicit SGD on Cox proportional hazards model. The different curves correspond to the 5%, 50%, and 95% quantiles of the MSE for 50 replications of implicit SGD iterates  $\theta_{2000}^{\text{im}}$ . The top and bottom dashed horizontal lines correspond, respectively, to the 95% quantile and the 5% quantile over the same 50 replications of the best possible MSE.

region of optimal MLE in a few thousands of iterations. In experiments with explicit SGD we were not able to replicate this performance because of numerical instability. To our best knowledge, there are no standard implementations of explicit SGD for estimating Cox proportional hazards models.

### 5.2.5 M-estimation

In this experiment we test the performance of implicit SGD, in particular Algorithm 5, on a M-estimation problem in a setup that is similar to the simulation example of [Donoho and Montanari \(2013, Example 2.4\)](#).

We set  $N = 1000$  data points and  $p = 200$  as the parameter dimension. We sample  $\theta_*$  as a random vector with norm  $\|\theta_*\| = 6\sqrt{p}$ , and sample the design matrix as  $X \sim \mathcal{N}_p(0, (1/N)I)$ . The outcomes are sampled i.i.d. from a “contaminated” normal distribution, i.e., with probability 95%,  $Y_n \sim \mathcal{N}(X_n^\top \theta_*, 1)$ , and  $Y_n = 10$  with probability 5%.

The results over 2000 iterations of implicit SGD are shown in Figure 4. In the figure we plot the 5% and 95% quantiles of MSE of implicit SGD over 100 replications of the experiment. We also plot (horizontal dashed line) the median of the MSE of the MLE estimator, computed using the `coxph` built-in command

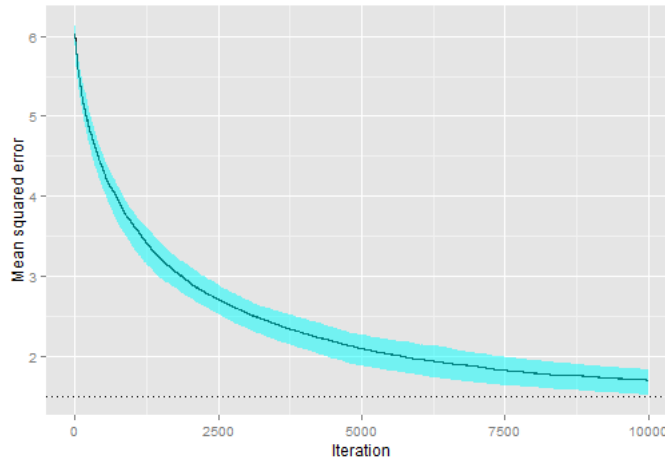


Figure 4: Implicit SGD on M-estimation task. Top curve corresponds to 95% percentile of MSE, middle curve is median, and bottom curve is 5% of MSE over 100 replications of implicit SGD iterates  $\theta_{2000}^{\text{im}}$ . The dashed horizontal line corresponds to the median MSE of maximum-likelihood over the same 100 replications.

of R.<sup>12</sup> We observe that SGD converges steadily to the best possible MSE. Similar behavior was observed under various modifications of the simulation parameters.

### 5.2.6 National Morbidity-Mortality Air Pollution study

The National Morbidity and Mortality Air Pollution (NMMAPS) study (Samet et al., 2000; Dominici et al., 2002) analyzed the risks of air pollution to public health. Several cities (108 in the US) are included in the study with daily measurements covering more than 13 years (roughly 5,000 days) including air pollution data (e.g. concentration of CO in the atmosphere) together with health outcome variables such as number of respiratory-related deaths.

The original study fitted a Poisson generalized additive model (GAM), separately for each city due to data set size. Recent research (Wood et al., 2014) has developed procedures similar to `biglm`'s iterative QR decomposition to fit all cities simultaneously on the full data set with approximately  $N = 1.2$  million observations and  $p = 802$  covariates (7 Gb in size). In this experiment, we construct a GAM model using data from all cities in the NMMAPS study in a process that is very similar (but not identical) to the data set of Wood et al. (2014).

<sup>12</sup>Documentation is available at <https://stat.ethz.ch/R-manual/R-devel/library/survival/html/coxph.html>.

Our final data set has  $N = 1,426,806$  observations and  $p = 794$  covariates including all cities in the NMMAPS study (8.6GB in size), and is fit using the simplest first-order implicit SGD procedure, i.e.,  $C_n = I$  and  $\gamma_1 = 1$ . The time to fit the entire model with implicit SGD was roughly 123.4 seconds, which is almost 6x faster than the time reported by [Wood et al. \(2014\)](#) of about 12 minutes on a similar home computer. We cannot directly compare the estimates from the two procedures because different versions of the data sets were used. However, we can compare the estimates of our model with the estimates of `glm()` on a random small subset of the data. In particular, we subsampled  $N = 50,000$  observations and  $p = 50$  covariates (19.5MB in size) and fit the smaller data set using implicit SGD and `glm()`. A Q-Q plot of the estimates is shown in Figure 5. We observe that the estimates of the SGD procedure are very close to MLE. Further replications of the aforementioned testing process revealed the same pattern, indicating that implicit SGD converged on all replications.

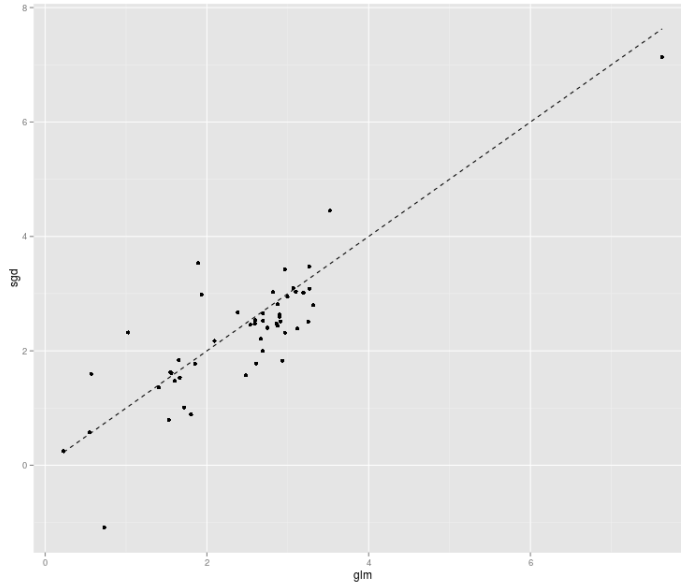


Figure 5: Estimates of implicit SGD (y-axis) and `glm()` (x-axis) on a subset of the NMMAPS data set with  $N = 50,000$  observations and  $p = 50$  covariates which is, roughly, 5% of the entire data set.

## 6 Discussion

Our theory in Section 3 suggests that implicit SGD is numerically stable and has known asymptotic variance and asymptotic distribution. Our experiments

in Section 5 showed that the empirical properties of SGD are well predicted by theory. In contrast, explicit SGD is unstable and cannot work well without problem-specific tuning. Thus, implicit SGD is a principled estimation procedure that is superior to classical explicit SGD.

Intuitively, implicit SGD leverages second-order information at every iteration, although this quantity is not explicitly computed in Eq. (4). To see this, assume both explicit and implicit SGD are at the same estimate  $\theta_0$ . Then, using definitions (3) and (4), a Taylor approximation of the gradient  $\nabla \log f(Y_n; X_n, \theta_n^{\text{im}})$  yields

$$\Delta \theta_n^{\text{im}} \approx [I + \gamma_n \hat{\mathcal{I}}(\theta_0; X_n, Y_n)]^{-1} \Delta \theta_n^{\text{sgd}}, \quad (27)$$

where  $\Delta \theta_n^{\text{im}} = \theta_n^{\text{im}} - \theta_0$  and  $\Delta \theta_n^{\text{sgd}} = \theta_n^{\text{sgd}} - \theta_0$ , and the matrix  $\hat{\mathcal{I}}(\theta_0; X_n, Y_n) = -\nabla^2 \log f(Y_n; X_n, \theta)|_{\theta=\theta_0}$  is the *observed* Fisher information at  $\theta_0$ . In other words, the implicit procedure is a *shrunked* version of the explicit one, where the shrinkage factor depends on the observed information.

Naturally, the implicit SGD iterate  $\theta_n^{\text{im}}$  has also a Bayesian interpretation. In particular,  $\theta_n^{\text{im}}$  is the posterior mode of a Bayesian model defined as

$$\begin{aligned} \theta | \theta_{n-1}^{\text{im}} &\sim \mathcal{N}(\theta_{n-1}^{\text{im}}, \gamma_n C_n) \\ Y_n | X_n, \theta &\sim f(\cdot; X_n, \theta). \end{aligned} \quad (28)$$

The explicit SGD update  $\theta_n^{\text{sgd}}$  can be written as in Eq.(28), however  $f$  needs to be substituted with its linear approximation around  $\theta_{n-1}^{\text{sgd}}$ . Thus, Eq. (28) provides an alternative explanation why implicit SGD is more principled than explicit SGD. Furthermore, it indicates possible improvements for implicit SGD. For example, the prior in Eq. (28) should be chosen to fit better the parameter space (e.g.,  $\theta_*$  being on the simplex). [Krakowski et al. \(2007\)](#) and [Nemirovski et al. \(2009\)](#) have argued that appropriate implicit updates can fit better in the geometry of the parameter space, and thus converge faster. Setting up the parameters of the prior is also crucial. Whereas in explicit SGD there is no statistical intuition behind learning rates  $\gamma_n$ , Eq. (28) reveals that in implicit SGD the terms  $(\gamma_n C_n)^{-1}$  encode the statistical information up to iteration  $n$ . It follows immediately that it is optimal, in general, to set  $\gamma_n C_n = \mathcal{I}(\theta_*)^{-1}/n$ , which is a special case of Theorem 3.2. Adaptive stochastic approximation methods in recursive estimation, typically being second-order as in Section 3.2.1, try to iteratively estimate the Fisher matrix  $\mathcal{I}(\theta_*)$  along with estimation of  $\theta_*$ .

The Bayesian formulation of Eq.(28) also explains the stability of implicit SGD. In Theorem 3.1 we showed that the initial conditions are discounted at an exponential rate, regardless of misspecification of the learning rates. This stability of implicit SGD allows several ideas for improvements. For example, constant learning rates could be used in implicit SGD to speed up convergence towards

a region around  $\theta_*$ . A sequential hypothesis test could decide on whether  $\theta_n^{\text{im}}$  has reached that region or not, and switch to the theoretically optimal  $1/n$  rate accordingly. Alternatively, we could run implicit SGD with AdaGrad learning rates (see Eq.(11)) and switch to  $1/n$  rates when the theoretical  $O(1/\sqrt{n})$  variance of AdaGrad becomes larger than the  $O(1/n)$  variance of implicit SGD. Such schemes using constant rates with explicit SGD are very hard to do in practice because of instability.

Regarding statistical efficiency, a key technical result in this paper is that the asymptotic variance of implicit SGD can be quantified exactly using Theorem 3.2. Optimal learning rates were suggested in Eq. (9) that depend on the eigenvalues of the unknown Fisher matrix  $\mathcal{I}(\theta_*)$ . In this paper, we used second-order procedures of Section 3.2.1 to iteratively estimate the eigenvalues, however better methods are certainly possible and could improve the performance of implicit SGD. For example, it is known that typical iterative methods (as in Section 3.2.1) usually overestimate the largest eigenvalue and underestimate the smallest eigenvalue, in small-to-moderate samples. This crucially affects the behavior of stochastic approximations with learning rates that depend on sample eigenvalues. Empirical Bayes methods have been shown to be superior in iterative estimation of eigenvalues of large matrices and it would be interesting to apply such methods to design the learning rates of implicit SGD procedures (Haff, 1980; Dey, 1988; Ahmed, 1998; Mestre, 2008).

Regarding computational efficiency, we developed Algorithm 1 which implements implicit SGD on a large family of statistical models. However, the trick used in fitting the Cox proportional hazards model in Section 4.3 can be more generally applied to models outside this family. For example, assume a log-likelihood gradient of the form  $s(X^\top \theta; Y)G(\theta; X, Y)$ , where both its scale  $s(\cdot)$  and direction  $G(\cdot)$  depend on model parameters  $\theta$ ; This violates conditions of Assumption 3.1(b). The implicit update (4) – where we set  $C_n = I$ , for simplicity – would be  $\theta_n^{\text{im}} = \theta_{n-1}^{\text{im}} + \gamma_n s(X_n^\top \theta_n^{\text{im}}; Y_n)G(\theta_n^{\text{im}}; X_n, Y_n)$ , which cannot be computed by Algorithm 1. One way to circumvent this problem is to use an implicit update only on the scale and use an explicit update on the direction, i.e.,  $\theta_n^{\text{im}} = \theta_{n-1}^{\text{im}} + \gamma_n s(X_n^\top \theta_n^{\text{im}}; Y_n)G(\theta_{n-1}^{\text{im}}; X_n, Y_n)$ . This form of updates expands the applicability of implicit SGD.

Finally, hypothesis testing and construction of confidence intervals using SGD estimates is an important issue that has hitherto remained unexplored. In experiments of Section 5.1.2 we showed that implicit SGD is indeed asymptotically normal in several reasonable simulation scenarios. However, as SGD procedures are iterative, there needs to be a rigorous and general method to decide whether SGD iterates have converged to the asymptotic regime. Several methods, such as bootstrapping the data set, could be used for that. Furthermore, conservative confidence intervals could be constructed through multivariate Chebyshev

inequalities or other strategies (Marshall and Olkin, 1960).

## 7 Conclusion

In this paper, we introduced a new stochastic gradient descent procedure that uses implicit updates (i.e., solves fixed-point equations) at every iteration, which we termed implicit SGD. Equation (27) shows, intuitively, that the iterates of implicit SGD are a shrunked version of the standard iterates, where the shrinkage factor depends on the observed Fisher information matrix. Thus, implicit SGD combines the computational efficiency of first-order methods with the numerical stability of second-order methods.

In a theoretical analysis of implicit SGD, we derived non-asymptotic upper bounds for the mean-squared errors of its iterates, and their asymptotic variance and normality. Our analysis suggests principled strategies to calibrate a hyper-parameter that is common to both explicit and implicit SGD procedures, known as the learning rate. We illustrated the use of implicit SGD for statistical estimation in generalized linear models, Cox proportional hazards model, and general M-estimation problems. Implicit SGD can also be applied on several other models, such as support vector machines or generalized additive models, with minimal modifications.

Viewed as statistical estimation procedures, our results suggest that implicit SGD has the same asymptotic efficiency to explicit SGD. However, the implicit procedure is significantly more stable than the explicit one with respect to misspecification of the learning rate. In general, explicit SGD procedures are sensitive to outliers and to misspecification of the learning rates, making it impossible to apply without problem-specific tuning. In theory and in extensive experiments, implicit procedures emerge as principled iterative estimation methods because they are numerically stable, they are robust to tuning of hyper-parameters, and their standard errors are well-predicted by theory. Thus, implicit stochastic gradient descent is poised to become a workhorse of estimation with large data sets in statistical practice.

## References

- SE Ahmed. Large-sample estimation strategies for eigenvalues of a wishart matrix. *Metrika*, 47(1):35–45, 1998.
- Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

- Shun-Ichi Amari, Hyeyoung Park, and Kenji Fukumizu. Adaptive method of realizing natural gradient learning for multilayer perceptrons. *Neural Computation*, 12(6):1399–1409, 2000.
- Francis Bach and Eric Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate  $o(1/n)$ . In *Advances in Neural Information Processing Systems*, pages 773–781, 2013.
- Sivaraman Balakrishnan, Martin J Wainwright, and Bin Yu. Statistical guarantees for the em algorithm: From population to sample-based analysis. *arXiv preprint arXiv:1408.2156*, 2014.
- JA Bather. *Stochastic approximation: A generalisation of the Robbins-Monro procedure*, volume 89. Mathematical Sciences Institute, Cornell University, 1989.
- Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.
- Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.
- Albert Benveniste, Pierre Priouret, and Michel Métivier. Adaptive algorithms and stochastic approximations. 1990.
- Dimitri P Bertsekas. Incremental proximal methods for large scale convex optimization. *Mathematical programming*, 129(2):163–195, 2011.
- Jock Blackard. *Comparison of neural networks and discriminant analysis in predicting forest cover types*. PhD thesis, Department of Forest Sciences, Colorado State University, 1998.
- Antoine Bordes, Léon Bottou, and Patrick Gallinari. Sgd-qn: Careful quasi-newton stochastic gradient descent. *The Journal of Machine Learning Research*, 10:1737–1754, 2009.
- Vivek S Borkar. Stochastic approximation. *Cambridge Books*, 2008.
- Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT’2010*, pages 177–186. Springer, 2010.
- Leon Bottou. Stochastic Gradient Descent Tricks. In *Neural Networks: Tricks of the Trade*, volume 1, pages 421–436. 2012.

- Richard H Byrd, SL Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-newton method for large-scale optimization. *arXiv preprint arXiv:1401.7020*, 2014.
- Olivier Cappé and Eric Moulines. On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):593–613, 2009.
- Han-Fu Chen, Lei Guo, and Ai-Jun Gao. Convergence and robustness of the robbins-monro algorithm truncated at randomly varying bounds. *Stochastic Processes and their Applications*, 27:217–231, 1987.
- Kai Lai Chung. On a stochastic approximation method. *The Annals of Mathematical Statistics*, pages 463–483, 1954.
- David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 187–220, 1972.
- Anthony Christopher Davison. *Statistical models*, volume 11. Cambridge University Press, 2003.
- Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, et al. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems*, pages 1223–1231, 2012.
- A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39: 1–38, 1977.
- Dipak K Dey. Simultaneous estimation of eigenvalues. *Annals of the Institute of Statistical Mathematics*, 40(1):137–147, 1988.
- Annette J Dobson and Adrian Barnett. *An introduction to generalized linear models*. CRC press, 2008.
- Francesca Dominici, Michael Daniels, Scott L Zeger, and Jonathan M Samet. Air pollution and mortality: estimating regional and national dose-response relationships. *Journal of the American Statistical Association*, 97(457):100–111, 2002.
- David Donoho and Andrea Montanari. High dimensional robust m-estimation: Asymptotic variance via approximate message passing. *arXiv preprint arXiv:1310.7320*, 2013.



- John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 999999:2121–2159, 2011a.
- John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011b.
- Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *The Journal of Machine Learning Research*, 11:625–660, 2010.
- Vaclav Fabian. On asymptotic normality in stochastic approximation. *The Annals of Mathematical Statistics*, pages 1327–1332, 1968a.
- Vaclav Fabian. On asymptotic normality in stochastic approximation. *The Annals of Mathematical Statistics*, pages 1327–1332, 1968b.
- Vaclav Fabian. On asymptotically efficient recursive estimation. *The Annals of Statistics*, pages 854–866, 1978.
- R. A. Fisher. *Statistical Methods for Research Workers*. Oliver and Boyd, Edinburgh, 1925.
- Ronald A Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 222:309–368, 1922.
- Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.
- Abraham P George and Warren B Powell. Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. *Machine learning*, 65(1):167–198, 2006.
- Peter J Green. Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 149–192, 1984.
- LR Haff. Empirical bayes estimation of the multivariate normal covariance matrix. *The Annals of Statistics*, pages 586–597, 1980.
- T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, second edition, 2011.

- Trevor J Hastie and Robert J Tibshirani. *Generalized additive models*, volume 43. CRC Press, 1990.
- Philipp Hennig and Martin Kiefel. Quasi-newton methods: A new direction. *The Journal of Machine Learning Research*, 14(1):843–865, 2013.
- Joe D Hoffman and Steven Frankel. *Numerical methods for engineers and scientists*. CRC press, 2001.
- Peter J Huber. *Robust statistics*. Springer, 2011.
- Peter J Huber et al. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.
- Prateek Jain, Ambuj Tewari, and Purushottam Kar. On iterative hard thresholding methods for high-dimensional m-estimation. In *Advances in Neural Information Processing Systems*, pages 685–693, 2014.
- Nikos Karampatziakis and John Langford. Online importance weight aware updates. *arXiv preprint arXiv:1011.1576*, 2010.
- Nouredine El Karoui. Spectrum estimation for large dimensional covariance matrices using random matrix theory. *The Annals of Statistics*, pages 2757–2790, 2008.
- Carl T Kelley. *Iterative methods for optimization*, volume 18. Siam, 1999.
- Jyrki Kivinen, Manfred K Warmuth, and Babak Hassibi. The p-norm generalization of the lms algorithm for adaptive filtering. *Signal Processing, IEEE Transactions on*, 54(5):1782–1793, 2006.
- John P Klein and Melvin L Moeschberger. *Survival analysis: techniques for censored and truncated data*. Springer Science & Business Media, 2003.
- Krzysztof A Krakowski, Robert E Mahony, Robert C Williamson, and Manfred K Warmuth. A geometric view of non-linear on-line stochastic gradient descent. *Author website*, 2007.
- Brian Kulis and Peter L Bartlett. Implicit online learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 575–582, 2010.
- Kenneth Lange. *Numerical analysis for statisticians*. Springer, 2010.

- Yann Le Cun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of IEEE*, 86(11):2278–2324, 1998.
- Erich Leo Lehmann and George Casella. *Theory of point estimation*, volume 31. Springer Science & Business Media, 1998.
- David Lewis, Yiming Yang, Tony Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5:361–397, 2004.
- Yuanqing Lin, Fengjun Lv, Shenghuo Zhu, Ming Yang, Timothee Cour, Kai Yu, Liangliang Cao, and Thomas Huang. Large-scale image classification: fast feature extraction and svm training. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1689–1696. IEEE, 2011.
- Pierre-Louis Lions and Bertrand Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.
- Lennart Ljung, Georg Pflug, and Harro Walk. *Stochastic approximation and optimization of random systems*, volume 17. Springer, 1992.
- Aleksandr Mikhailovich Lyapunov. The general problem of the stability of motion. *International Journal of Control*, 55(3):531–534, 1992.
- Albert W Marshall and Ingram Olkin. Multivariate chebyshev inequalities. *The Annals of Mathematical Statistics*, pages 1001–1014, 1960.
- R Douglas Martin and C Johan Masreliez. Robust estimation via stochastic approximation. *Information Theory, IEEE Transactions on*, 21(3):263–271, 1975.
- Xavier Mestre. Improved estimation of eigenvalues and eigenvectors of covariance matrices using their sample estimates. *Information Theory, IEEE Transactions on*, 54(11):5113–5129, 2008.
- Alan J Miller. Algorithm as 274: Least squares routines to supplement those of gentleman. *Applied Statistics*, pages 458–478, 1992.
- Eric Moulines and Francis R Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, pages 451–459, 2011.

- Jin-Ichi Nagumo and Atsuhiko Noda. A learning method for system identification. *Automatic Control, IEEE Transactions on*, 12(3):282–287, 1967.
- National Research Council. *Frontiers in Massive Data Analysis*. The National Academies Press, Washington, DC, 2013.
- J.A. Nelder and R.W.M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, pages 370–384, 1972.
- Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.
- DB Nemirovski, Yudin. *Problem complexity and method efficiency in optimization*. Wiley (Chichester and New York), 1983.
- Mikhail Borisovich Nevelson and Rafail Zalmanovich Khasminskii. *Stochastic approximation and recursive estimation*, volume 47. Amer Mathematical Society, 1973.
- Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):123–231, 2013.
- BT Poljak and Ja Z Tsypkin. Robust identification. *Automatica*, 16(1):53–63, 1980.
- Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992a.
- Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992b.
- Boris Teodorovich Polyak and Ya Z Tsypkin. Adaptive estimation algorithms: convergence, optimality, stability. *Avtomatika i Telemekhanika*, (3):71–84, 1979.
- Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.
- R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

- Lorenzo Rosasco, Silvia Villa, and Bang Công Vũ. Convergence of stochastic proximal gradient algorithm. *arXiv preprint arXiv:1403.5074*, 2014.
- David Ruppert. Efficient estimations from a slowly convergent robbins-monro process. Technical report, Cornell University Operations Research and Industrial Engineering, 1988.
- Jerome Sacks. Asymptotic distribution of stochastic approximation procedures. *The Annals of Mathematical Statistics*, 29(2):373–405, 1958.
- David J Sakrison. Efficient recursive estimation; application to estimating the parameters of a covariance function. *International Journal of Engineering Science*, 3(4):461–483, 1965.
- Jonathan M Samet, Scott L Zeger, Francesca Dominici, Frank Curriero, Ivan Coursac, Douglas W Dockery, Joel Schwartz, and Antonella Zanobetti. The national morbidity, mortality, and air pollution study. *Part II: morbidity and mortality from air pollution in the United States Res Rep Health Eff Inst*, 94 (pt 2):5–79, 2000.
- Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. Technical report, HAL 00860051, 2013.
- Li Cheng SVN Schuurmans and SW Caelli. Implicit online learning with kernels. *Advances in neural information processing systems*, 19:249, 2007.
- Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. *arXiv preprint arXiv:1212.1824*, 2012.
- Noah Simon, Jerome Friedman, Trevor Hastie, Rob Tibshirani, et al. Regularization paths for coxs proportional hazards model via coordinate descent. *Journal of statistical software*, 39(5):1–13, 2011.
- Yoram Singer and John C Duchi. Efficient learning using forward-backward splitting. In *Advances in Neural Information Processing Systems*, pages 495–503, 2009.
- Dirk TM Slock. On the convergence behavior of the lms and the normalized lms algorithms. *Signal Processing, IEEE Transactions on*, 41(9):2811–2825, 1993.
- Soeren Sonnenburg, Vojtech Franc, Elad Yom-Tov, and Michele Sebag. Pascal large scale learning challenge, 2008.

- P. Toulis, E. Airoldi, and J. Rennie. Statistical analysis of stochastic gradient methods for generalized linear models. *JMLR W&CP*, 32(1):667–675, 2014.
- Panos Toulis and Edoardo M. Airoldi. Scalable estimation strategies based on stochastic approximations: classical results and new insights. *Statistics and Computing*, 25(4):781–795, 2015. ISSN 0960-3174. doi: 10.1007/s11222-015-9560-y. URL <http://dx.doi.org/10.1007/s11222-015-9560-y>.
- Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.
- Bernard Widrow and Marcian E Hoff. Adaptive switching circuits. *Defense Technical Information Center*, 1960.
- Simon N Wood, Yannig Goude, and Simon Shaw. Generalized additive models for large data sets. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 2014.
- Lin Xiao and Tony Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24:2057–2075, 2014.
- Wei Xu. Towards optimal one pass large scale learning with averaged stochastic gradient descent. *arXiv preprint arXiv:1107.2490*, 2011.
- Tong Zhang. Solving large scale linear prediction problems using gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, page 116. ACM, 2004.

## A R code

All experiments were run using our `sgd` R package that implements explicit SGD (3) and implicit SGD (4). The package is published at CRAN, <http://cran.r-project.org/web/packages/sgd/index.html>, and has been co-authored with Dustin Tran and Kuang Ye. The repository at <https://github.com/ptoulis/implicit-glms> contains legacy R code for explicit and implicit SGD. The folder `examples/aos2014` in that repository contains standalone code for the experiments in this paper, but it has been superceded by the `sgd` R package.

## B Useful lemmas

In this section, we will prove certain lemmas on recursions that will be useful for the non-asymptotic analysis of the implicit procedures. All following results are stated under a combination of Assumptions 3.1.

**Lemma B.1.** *Consider a sequence  $b_n$  such that  $b_n \downarrow 0$  and  $\sum_{i=1}^{\infty} b_i = \infty$ . Then, there exists a positive constant  $K > 0$ , such that*

$$\prod_{i=1}^n \frac{1}{1 + b_i} \leq \exp(-K \sum_{i=1}^n b_i). \quad (29)$$

*Proof.* The function  $x \log(1 + 1/x)$  is increasing-concave in  $(0, \infty)$ . From  $b_n \downarrow 0$  it follows that  $\log(1 + b_n)/b_n$  is non-increasing. Consider the value  $K \stackrel{\text{def}}{=} \log(1 + b_1)/b_1$ . Then,  $(1 + b_n)^{-1} \leq \exp(-K b_n)$ . Successive applications of this inequality yields Ineq. (29). The upper bound is more accurate when  $b_n$  takes lower values.  $\square$

**Lemma B.2.** *Consider sequences  $a_n \downarrow 0$ ,  $b_n \downarrow 0$ , and  $c_n \downarrow 0$  such that,  $a_n = o(b_n)$ ,  $\sum_{i=1}^{\infty} a_i \stackrel{\text{def}}{=} A < \infty$ , and there is  $n'$  such that  $c_n/b_n < 1$  for all  $n > n'$ . Define,*

$$\delta_n \stackrel{\text{def}}{=} \frac{1}{a_n} (a_{n-1}/b_{n-1} - a_n/b_n) \text{ and } \zeta_n \stackrel{\text{def}}{=} \frac{c_n}{b_{n-1}} \frac{a_{n-1}}{a_n}, \quad (30)$$

*and suppose that  $\delta_n \downarrow 0$  and  $\zeta_n \downarrow 0$ . Pick a positive  $n_0$  such that  $\delta_n + \zeta_n < 1$  and  $(1 + c_n)/(1 + b_n) < 1$ , for all  $n \geq n_0$ .*

*Consider a positive sequence  $y_n > 0$  that satisfies the recursive inequality,*

$$y_n \leq \frac{1 + c_n}{1 + b_n} y_{n-1} + a_n. \quad (31)$$

*Then, for every  $n > 0$ ,*

$$y_n \leq K_0 \frac{a_n}{b_n} + Q_1^n y_0 + Q_{n_0+1}^n (1 + c_1)^{n_0} A, \quad (32)$$

where  $K_0 \stackrel{\text{def}}{=} (1 + b_1)(1 - \delta_{n_0} - \zeta_{n_0})^{-1}$ , and  $Q_i^n = \prod_{j=i}^n (1 + c_j)/(1 + b_i)$ , such that  $Q_i^n = 1$  if  $n < i$ , by definition.

*Proof.* We consider two separate cases, namely,  $n < n_0$  and  $n \geq n_0$ , and then we will combine the respective bounds.

**Analysis for  $n < n_0$ .** We first find a crude bound for  $Q_{i+1}^n$ . It holds,

$$Q_{i+1}^n \leq (1 + c_{i+1})(1 + c_{i+2}) \cdots (1 + c_n) \leq (1 + c_1)^{n_0}, \quad (33)$$

since  $c_1 \geq c_n$  ( $c_n \downarrow 0$  by definition) and there are no more than  $n_0$  terms in the product. From Ineq. (31) we get

$$\begin{aligned} y_n &\leq Q_1^n y_0 + \sum_{i=1}^n Q_{i+1}^n a_i \quad [\text{by expanding recursive Ineq. (31)}] \\ &\leq Q_1^n y_0 + (1 + c_1)^{n_0} \sum_{i=1}^n a_i \quad [\text{using Ineq. (33)}] \\ &\leq Q_1^n y_0 + (1 + c_1)^{n_0} A. \end{aligned} \quad (34)$$

This inequality holds also for  $n = n_0$ .

**Analysis for  $n \geq n_0$ .** In this case, we have for all  $n \geq n_0$ ,

$$\begin{aligned} (1 + b_1)(1 - \delta_n - \zeta_n)^{-1} &\leq K_0 \quad [\text{by definition of } n_0, K_0] \\ K_0(\delta_n + \zeta_n) + 1 + b_1 &\leq K_0 \\ K_0(\delta_n + \zeta_n) + 1 + b_n &\leq K_0 \quad [\text{because } b_n \leq b_1, \text{ since } b_n \downarrow 0] \\ \frac{1}{a_n} K_0 \left( \frac{a_{n-1}}{b_{n-1}} - \frac{a_n}{b_n} \right) + \frac{1}{a_n} K_0 \frac{c_n a_{n-1}}{b_{n-1}} + 1 + b_n &\leq K_0 \quad [\text{by definition of } \delta_n, \zeta_n] \\ a_n(1 + b_n) &\leq K_0 a_n - K_0 \left( \frac{(1 + c_n)a_{n-1}}{b_{n-1}} - \frac{a_n}{b_n} \right) \\ a_n &\leq K_0 \left( \frac{a_n}{b_n} - \frac{1 + c_n}{1 + b_n} \frac{a_{n-1}}{b_{n-1}} \right). \end{aligned} \quad (35)$$

Now we combine Ineqs. (35) and (31) to obtain

$$(y_n - K_0 \frac{a_n}{b_n}) \leq \frac{1 + c_n}{1 + b_n} (y_{n-1} - K_0 \frac{a_{n-1}}{b_{n-1}}). \quad (36)$$

For brevity, define  $s_n \stackrel{\text{def}}{=} y_n - K_0 a_n / b_n$ . Then, from Ineq. (36),  $s_n \leq \frac{1 + c_n}{1 + b_n} s_{n-1}$ , where  $\frac{1 + c_n}{1 + b_n} < 1$  since  $n \geq n_0$ . Assume  $n_1$  is the smallest integer such that  $n_1 \geq n_0$  and  $s_{n_1} \leq 0$  (existence of  $n_1$  is not crucial.) For all  $n \geq n_1$ , it follows  $s_n \leq 0$ , and



thus  $y_n \leq K_0 a_n / b_n$  for all  $n \geq n_1$ . Alternatively, when  $n_0 \leq n < n_1$ , all  $s_n$  are positive. Using Ineq. (36) we have  $s_n \leq (\prod_{i=n_0+1}^n \frac{1+c_i}{1+b_i}) s_{n_0} \stackrel{\text{def}}{=} Q_{n_0+1}^n s_{n_0}$ , and thus

$$\begin{aligned} y_n - K_0 \frac{a_n}{b_n} &\leq Q_{n_0+1}^n s_{n_0} \quad [\text{by definition of } s_n] \\ y_n &\leq K_0 \frac{a_n}{b_n} + Q_{n_0+1}^n y_{n_0} \quad [\text{because } s_n \leq y_n] \\ y_n &\leq K_0 \frac{a_n}{b_n} + Q_1^n y_0 + Q_{n_0+1}^n (1 + c_1)^{n_0} A. \quad [\text{by Ineq. (34) on } y_{n_0}] \end{aligned} \quad (37)$$

Combining this result with Ineqs. (34) and (37), we obtain

$$y_n \leq K_0 \frac{a_n}{b_n} + Q_1^n y_0 + Q_{n_0+1}^n (1 + c_1)^{n_0} A, \quad (38)$$

since  $Q_i^n = 1$  for  $n < i$ , by definition.  $\square$

**Corollary B.1.** *In Lemma B.2 assume  $a_n = a_1 n^{-\alpha}$  and  $b_n = b_1 n^{-\beta}$ , and  $c_n = 0$ , where  $\alpha > \beta$ , and  $a_1, b_1, \beta > 0$  and  $\alpha > 1$ . Then,*

$$y_n \leq 2 \frac{a_1(1 + b_1)}{b_1} n^{-\alpha+\beta} + \exp(-\log(1 + b_1)n^{1-\beta})[y_0 + (1 + b_1)^{n_0} A], \quad (39)$$

where  $n_0 > 0$  and  $A = \sum_i a_i < \infty$ .

*Proof.* In this proof, we will assume that the inequalities  $(n-1)^{-\gamma} - n^{-\gamma} \leq n^{-1-\gamma}$  and  $\sum_{i=1}^n i^{-\gamma} \geq n^{1-\gamma}$  hold for  $n \geq n'$ , where  $0 < \gamma < 1$ , and  $n'$  is an appropriate threshold, that is generally small (e.g.,  $n' = 14$  if  $\gamma = 0.1$ ,  $n' = 5$  if  $\gamma = 0.5$ , and  $n' = 9$  if  $\gamma = 0.9$ .)

By definition,

$$\delta_n \stackrel{\text{def}}{=} \frac{1}{a_n} \left( \frac{a_{n-1}}{b_{n-1}} - \frac{a_n}{b_n} \right) = \frac{1}{a_1 n^{-\alpha}} \frac{a_1}{b_1} ((n-1)^{-\alpha+\beta} - n^{-\alpha+\beta}) \leq \frac{1}{b_1} n^{-1+\beta}. \quad (40)$$

Also,  $\zeta_n = 0$  since  $c_n = 0$ . We can take  $n_0 = \lceil (2/b_1)^{1/(1-\beta)} \rceil$ , for which  $\delta_{n_0} \leq 1/2$ . Therefore,  $K_0 \stackrel{\text{def}}{=} (1+b_1)(1-\delta_{n_0})^{-1} \leq 2(1+b_1)$ ; we can simply take  $K_0 = 2(1+b_1)$ . Since  $c_n = 0$ ,  $Q_i^n = \prod_{j=i}^n (1+b_j)^{-1}$ . Thus,

$$\begin{aligned} Q_1^n &\geq (1+b_1)^{-n}, \text{ and} \\ Q_1^n &\leq \exp(-\log(1+b_1)/b_1 \sum_{i=1}^n b_i), \quad [\text{by Lemma B.1.}] \\ Q_1^n &\leq \exp(-\log(1+b_1)n^{1-\beta}). \quad [\text{because } \sum_{i=1}^n i^{-\beta} \geq n^{1-\beta}.] \end{aligned} \quad (41)$$

Lemma B.2 and Ineqs. (41) imply

$$\begin{aligned}
y_n &\leq K_0 \frac{a_n}{b_n} + Q_1^n y_0 + Q_{n_0+1}^n (1 + c_1)^{n_0} A \quad [\text{by Lemma B.2}] \\
&\leq 2 \frac{a_1(1 + b_1)}{b_1} n^{-\alpha+\beta} + Q_1^n [y_0 + (1 + b_1)^{n_0} A] \quad [\text{by Ineqs. (41), } c_1 = 0] \\
&\leq 2 \frac{a_1(1 + b_1)}{b_1} n^{-\alpha+\beta} + \exp(-\log(1 + b_1)n^{1-\beta}) [y_0 + (1 + b_1)^{n_0} A], \quad (42)
\end{aligned}$$

where the last inequality also follows from Ineqs. (41).  $\square$

**Lemma B.3.** *Suppose Assumptions 3.1(b) (c), and (d) hold. Then, almost surely,*

$$\lambda_n \geq \frac{1}{1 + \gamma_n \underline{\lambda}_c \underline{\lambda}_f p}, \quad (43)$$

$$\|\theta_n^{\text{im}} - \theta_{n-1}^{\text{im}}\|^2 \leq 4L_0^2 \gamma_n^2, \quad (44)$$

where  $\lambda_n$  is defined in Theorem 4.1, and  $\theta_n^{\text{im}}$  is the  $n$ th iterate of implicit SGD (4).

*Proof.* For the first part, from Theorem 4.1, the random variable  $\lambda_n$  satisfies

$$\ell(X_n^\top \theta_n^{\text{im}}; Y_n) = \lambda_n \ell(X_n^\top \theta_{n-1}^{\text{im}}; Y_n). \quad (45)$$

Using definition (4),

$$\theta_n^{\text{im}} = \theta_{n-1}^{\text{im}} + \gamma_n \lambda_n \ell(X_n^\top \theta_{n-1}^{\text{im}}; Y_n) C_n X_n. \quad (46)$$

We use this definition of  $\theta_n^{\text{im}}$  into Eq.(45) and perform a Taylor approximation on  $\ell$  to obtain

$$\ell(X_n^\top \theta_n^{\text{im}}; Y_n) = \ell(X_n^\top \theta_{n-1}^{\text{im}}; Y_n) + \tilde{\ell}'' \gamma_n \lambda_n X_n^\top C_n X_n, \quad (47)$$

where  $\tilde{\ell}'' = \ell''(\delta X_n^\top \theta_{n-1}^{\text{im}} + (1 - \delta) X_n^\top \theta_n^{\text{im}}; Y_n)$ , and  $\delta \in [0, 1]$ . Thus, by definition,

$$\begin{aligned}
(1 - \gamma_n \tilde{\ell}'' X_n^\top C_n X_n) \lambda_n &= 1 \quad [\text{using Eq.(47) and Eq.(45)}] \\
(1 - \gamma_n \underline{\lambda}_c \tilde{\ell}'' \|X_n\|^2) \lambda_n &\geq 1 \quad [\text{because } \ell'' < 0 \text{ and by Assumption 3.1(e)}] \\
(1 + \gamma_n \underline{\lambda}_c \text{trace}(\hat{\mathcal{I}}_n(\tilde{\theta}))) \lambda_n &\geq 1 \quad [\tilde{\theta} = \delta \theta_{n-1}^{\text{im}} + (1 - \delta) \theta_n^{\text{im}}] \\
(1 + \gamma_n \underline{\lambda}_c \underline{\lambda}_f p) \lambda_n &\geq 1 \quad [\text{by Assumption 3.1(d)}]
\end{aligned} \quad (48)$$

For the second part, since the log-likelihood is differentiable (Assumption 3.1(b)) we can re-write the definition of implicit SGD (4) as

$$\theta_n^{\text{im}} = \arg \max \left\{ -\frac{1}{2\gamma_n} \|\theta - \theta_{n-1}^{\text{im}}\|^2 + \ell(X_n^\top \theta; Y_n) \right\}.$$

Therefore, setting  $\theta \equiv \theta_{n-1}^{\text{im}}$  in the above equation,

$$\begin{aligned} -\frac{1}{2\gamma_n} \|\theta_n^{\text{im}} - \theta_{n-1}^{\text{im}}\|^2 + \ell(X_n^\top \theta_n^{\text{im}}; Y_n) &\geq \ell(X_n^\top \theta_{n-1}^{\text{im}}; Y_n) \\ \|\theta_n^{\text{im}} - \theta_{n-1}^{\text{im}}\|^2 &\leq 2\gamma_n (\ell(X_n^\top \theta_n^{\text{im}}; Y_n) - \ell(X_n^\top \theta_{n-1}^{\text{im}}; Y_n)) \\ \|\theta_n^{\text{im}} - \theta_{n-1}^{\text{im}}\|^2 &\leq 2\gamma_n L_0 \|\theta_n^{\text{im}} - \theta_{n-1}^{\text{im}}\|. \end{aligned} \quad (49)$$

□

## C Non-asymptotic analysis

**Theorem 3.1.** *Let  $\delta_n \triangleq \mathbb{E}(\|\theta_n^{\text{im}} - \theta_\star\|^2)$  and  $\kappa \triangleq 1 + \gamma_1 \underline{\lambda}_c \underline{\lambda}_f \mu_0$ , where  $\mu_0 \in [1/(1 + \gamma_1 \underline{\lambda}_f \underline{\lambda}_c (p-1)), 1]$ . Suppose that Assumptions 3.1(a),(b),(c), (d) and (e) hold. Then, there exists constant  $n_0$  such that,*

$$\delta_n \leq \frac{8L_0^2 \overline{\lambda}_c^2 \gamma_1 \kappa}{\underline{\lambda}_c \underline{\lambda}_f \mu_0} n^{-\gamma} + \exp(-\log \kappa \cdot n^{1-\gamma}) [\delta_0 + \kappa^{n_0} \Gamma^2],$$

where  $\Gamma^2 = 4L_0^2 \overline{\lambda}_c^2 \sum_i \gamma_i^2 < \infty$ , and  $n_0$  is defined in Corollary B.1.

*Proof.* Starting from Procedure (4) we have

$$\begin{aligned} \theta_n^{\text{im}} - \theta_\star &= \theta_{n-1}^{\text{im}} - \theta_\star + \gamma_n C_n \nabla \log f(Y_n; X_n, \theta_n^{\text{im}}) \\ \theta_n^{\text{im}} - \theta_\star &= \theta_{n-1}^{\text{im}} - \theta_\star + \gamma_n \lambda_n C_n \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{im}}) \quad [\text{By Theorem 4.1}] \\ \|\theta_n^{\text{im}} - \theta_\star\|^2 &= \|\theta_{n-1}^{\text{im}} - \theta_\star\|^2 \\ &\quad + 2\gamma_n \lambda_n (\theta_{n-1}^{\text{im}} - \theta_\star)^\top C_n \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{im}}) \\ &\quad + \gamma_n^2 \|C_n \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{im}})\|^2. \end{aligned} \quad (50)$$

The last term can be simply bounded since  $\nabla \log f(Y_n; X_n, \theta_n^{\text{im}}) = \theta_n^{\text{im}} - \theta_{n-1}^{\text{im}}$  by definition; thus,

$$\|C_n \nabla \log f(Y_n; X_n, \theta_n^{\text{im}})\|^2 \leq \overline{\lambda}_c^2 \|\theta_n^{\text{im}} - \theta_{n-1}^{\text{im}}\|^2 \leq 4L_0^2 \overline{\lambda}_c^2 \gamma_n^2, \quad (51)$$

which holds almost surely by Lemma B.3-Eq.(44).

For the second term, we have

$$\begin{aligned}
\mathbb{E}((\theta_{n-1}^{\text{im}} - \theta_*)^\top C_n \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{im}})) \\
&= \mathbb{E}((\theta_{n-1}^{\text{im}} - \theta_*)^\top C_n \nabla h(\theta_{n-1}^{\text{im}})) \\
&= \mathbb{E}((z_{n-1} - z_*)^\top \nabla h(z_{n-1})) \quad [\text{change of variables, } z_n \stackrel{\text{def}}{=} C_n^{1/2} \theta_n^{\text{im}}] \\
&\leq -\underline{\lambda}_f \mathbb{E}(\|z_{n-1} - z_*\|^2) \quad [\text{by strong convexity, Assumption 3.1(d).}] \\
&\leq -\frac{\underline{\lambda}_f \underline{\lambda}_c}{2} \mathbb{E}(\|\theta_{n-1}^{\text{im}} - \theta_*\|^2) \quad [\text{by change of variables, } z_n = C_n^{1/2} \theta_n^{\text{im}} \text{ and Assumption 3.1(e).}]
\end{aligned} \tag{52}$$

If we combine Lemma B.3 and Ineq.(52) we obtain

$$\mathbb{E}(\lambda_n (\theta_{n-1}^{\text{im}} - \theta_*)^\top C_n \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{im}}) | \mathcal{F}_{n-1}) \leq -\frac{\underline{\lambda}_f \underline{\lambda}_c}{2(1 + \gamma_n \underline{\lambda}_c \underline{\lambda}_f p)} \mathbb{E}(\|\theta_{n-1}^{\text{im}} - \theta_*\|^2). \tag{53}$$

Taking expectations in Eq. (50) and substituting Ineq. (51) and Ineq.(52) into Eq.(50) yields the recursion,

$$\mathbb{E}(\|\theta_n^{\text{im}} - \theta_*\|^2) \leq (1 - \frac{\gamma_n \underline{\lambda}_c \underline{\lambda}_f}{1 + \gamma_n \underline{\lambda}_c \underline{\lambda}_f p}) \mathbb{E}(\|\theta_{n-1}^{\text{im}} - \theta_*\|^2) + 4L_0^2 \bar{\lambda}_c^{-2} \gamma_n^2. \tag{54}$$

Through simple algebra,

$$(1 - \frac{\gamma_n \underline{\lambda}_c \underline{\lambda}_f}{1 + \gamma_n \underline{\lambda}_c \underline{\lambda}_f p}) \leq \frac{1}{1 + \gamma_n \mu}, \tag{55}$$

where  $\mu \leq \underline{\lambda}_c \underline{\lambda}_f / (1 + \gamma_n \underline{\lambda}_c \underline{\lambda}_f (p-1))$ . Depending on the desired accuracy of the bounds of Ineq.(54) we can pick  $\mu$  appropriately small. For example, if we want to solve the Ineq.(54) from  $n = 1$ , then we should take  $\mu = \underline{\lambda}_c \underline{\lambda}_f / (1 + \gamma_1 \underline{\lambda}_c \underline{\lambda}_f (p-1))$ ; this would result in the “worst-case” upper-bound. The best-case upper bound is obtained by setting  $\mu$  at its limit value when  $\gamma_n \rightarrow 0$ , i.e.,  $\mu = \underline{\lambda}_c \underline{\lambda}_f$ . Thus, we will assume  $\mu = \underline{\lambda}_c \underline{\lambda}_f \mu_0$ , where  $\mu_0 \in [1/(1 + \gamma_1 \underline{\lambda}_c \underline{\lambda}_f (p-1)), 1]$ . Then, we can write recursion (54) as

$$\mathbb{E}(\|\theta_n^{\text{im}} - \theta_*\|^2) \leq \frac{1}{1 + \gamma_n \underline{\lambda}_f \underline{\lambda}_c \mu_0} \mathbb{E}(\|\theta_{n-1}^{\text{im}} - \theta_*\|^2) + 4L_0^2 \bar{\lambda}_c^{-2} \gamma_n^2. \tag{56}$$

We can now apply Corollary B.1 with  $a_n \equiv 4L_0^2 \bar{\lambda}_c^{-2} \gamma_n^2$  and  $b_n \equiv \gamma_n \underline{\lambda}_f \underline{\lambda}_c \mu_0$ .  $\square$

**Note.** Assuming Lipschitz continuity of the gradient  $\nabla \ell$  instead of function  $\ell$  would not critically alter the main result of Theorem 3.1. In fact, assuming Lipschitz continuity with constant  $L$  of  $\nabla \ell$  and boundedness of  $\mathbb{E}(\|\nabla \log f(Y_n; X_n, \theta_*)\|^2) \leq \sigma^2$ , as it is typical in the literature, would simply add a term  $\gamma_n^2 L^2 \mathbb{E}(\|\theta_n^{\text{im}} - \theta_*\|^2) + \gamma_n^2 \sigma^2$  in the right-hand side of Eq.(50). In this case the upper-bound is always satisfied for  $n$  such that  $\gamma_n^2 L^2 > 1$ , which also highlights a difference of implicit SGD with explicit SGD, as in explicit SGD the term  $\gamma_n^2 L^2 \|\theta_{n-1}^{\text{sgd}} - \theta_*\|^2$  increases the upper bound and can make  $\|\theta_n^{\text{sgd}} - \theta_*\|^2$  diverge. For,  $\gamma_n^2 L^2 < 1$ , the discount factor for implicit SGD would be  $(1 - \gamma_n^2 L^2)^{-1}(1 + \gamma_n \lambda_f \lambda_c)^{-1}$ , which could then be bounded by a quantity  $(1 + \gamma_n c)^{-1}$  for some constant  $c$ . This would lead to a solution that is similar to Theorem 3.1.

## D Asymptotic analysis

Here we prove the main result on the asymptotic variance of implicit SGD. First, we introduce linear maps  $\mathbb{L}_B \{\cdot\}$  defined as  $\mathbb{L}_B \{X\} = \frac{1}{2}(BX + XB)$ , where  $B$  is symmetric positive-definite matrix and  $X$  is bounded. The identity map is denoted as  $\mathbb{I}$  and it holds  $\mathbb{I} \{X\} = X$  for all  $X$ . Also,  $\mathbb{L}_0$  is the null operator for which  $\mathbb{L}_0 \{X\} = 0$  for all  $X$ . By the Lyapunov theorem (Lyapunov, 1992) the map  $\mathbb{L}_B$  is one-to-one and thus the inverse operator  $\mathbb{L}_B^{-1} \{\cdot\}$  is well-defined. Furthermore, we define the norm of a linear map as  $\|\mathbb{L}_B\| = \max_{\|X\|=1} \|\mathbb{L}_B \{X\}\|$ . For bounded inputs  $X$ , it holds  $\|\mathbb{L}_B\| = O(\|B\|)$ .

**Lemma D.1.** *Suppose that the sequence  $\{\gamma_n\}$  satisfies Assumption 3.1(a). Consider the matrix recursions*

$$X_n = \mathbb{L}_{I - \gamma_n B_n} \{X_{n-1}\} + \gamma_n(C + D_n), \quad (57)$$

$$Y_n = \mathbb{L}_{I + \gamma_n B_n}^{-1} \{X_{n-1} + \gamma_n(C + D_n)\}, \quad (58)$$

such that

- (a) All matrices  $X_n, Y_n, B_n, D_n$  and  $C$  are bounded,
- (b)  $B_n \rightarrow B$  is positive-definite and  $\|B_n - B_{n-1}\| = O(\gamma_n^2)$ ,
- (c)  $C$  is a fixed matrix and  $D_n \rightarrow 0$ .

Then, both recursions approximate the matrix  $\mathbb{L}_B^{-1} \{C\}$  i.e.,

$$\|X_n B + B X_n - 2C\| \rightarrow 0 \text{ and } \|Y_n B + B Y_n - 2C\| \rightarrow 0. \quad (59)$$

If, in addition,  $B$  and  $C$  commute then  $X_n, Y_n \rightarrow B^{-1}C$ .

*Proof.* We make the following definitions.

$$\Gamma_n \stackrel{\text{def}}{=} I - \gamma_n B_n, \quad (60)$$

$$P_i^n \stackrel{\text{def}}{=} \mathbb{L}_{\Gamma_n} \circ \mathbb{L}_{\Gamma_{n-1}} \circ \cdots \mathbb{L}_{\Gamma_i}, \quad (61)$$

where the symbol  $\circ$  denotes successive application of the linear maps, and  $P_i^n = \mathbb{I}$  if  $n < i$ , by definition. It follows,

$$\|P_i^n\| = O\left(\prod_{j=i}^n \|I - \gamma_j B_j\|\right) \leq K_0 e^{-K_1 \sum_{j=i}^n \gamma_j}, \quad (62)$$

for suitable constants  $K_0, K_1$  (see [Polyak and Juditsky, 1992a](#), Appendix, Part 3). Let  $\Gamma(n) = K_1 \sum_{i=1}^n \gamma_i$ . By Assumption 3.1(a),  $\Gamma(n) \rightarrow \infty$  and thus  $P_i^n \rightarrow \mathbb{L}_0$  as  $n \rightarrow \infty$  and  $i$  is fixed. The matrix recursion in Lemma D.1 can be rewritten as  $X_n = \mathbb{L}_{\Gamma_n} \{X_{n-1}\} + \gamma_n C + \gamma_n D_n$ . Solving the recursion yields

$$\begin{aligned} X_n &= \mathbb{L}_{\Gamma_n} \circ \mathbb{L}_{\Gamma_{n-1}} \circ \cdots \mathbb{L}_{\Gamma_1} \{X_0\} + \gamma_n C + \gamma_n D_n \\ &\quad + a_{n-1} \mathbb{L}_{\Gamma_n} \{C\} + a_{n-1} \mathbb{L}_{\Gamma_n} \{D_{n-1}\} \\ &\quad + \cdots + \\ &\quad + a_1 \mathbb{L}_{\Gamma_n} \circ \mathbb{L}_{\Gamma_{n-1}} \circ \cdots \mathbb{L}_{\Gamma_2} \{C\} + a_1 \mathbb{L}_{\Gamma_n} \circ \mathbb{L}_{\Gamma_{n-1}} \circ \cdots \mathbb{L}_{\Gamma_2} \{D_1\} \\ &\stackrel{\text{def}}{=} P_1^n \{X_0\} + S_n \{C\} + \tilde{D}_n, \end{aligned} \quad (63)$$

where we have defined the linear map  $S_n = \sum_{i=1}^n \gamma_i P_{i+1}^n$  and the matrix  $\tilde{D}_n = \sum_{i=1}^n \gamma_i P_{i+1}^n \{D_i\}$ . Since  $P_1^n \rightarrow \mathbb{L}_0$ , our goal is to prove that  $S_n \rightarrow \mathbb{L}_B^{-1}$  and  $\tilde{D}_n \rightarrow 0$ . By definition,

$$\sum_{i=1}^n \gamma_i P_{i+1}^n = \mathbb{L}_{B_n}^{-1} + \sum_{i=2}^n P_i^n (\mathbb{L}_{B_{i-1}}^{-1} - \mathbb{L}_{B_i}^{-1}) - P_1^n \mathbb{L}_{B_1}^{-1}. \quad (64)$$

To see this, first note that  $\gamma_n I = (I - \Gamma_n) B_n^{-1}$  for every  $n$ , and thus

$$\gamma_n \mathbb{I} = \mathbb{L}_{I - \Gamma_n} \circ \mathbb{L}_{B_n}^{-1}. \quad (65)$$

Therefore, if we collect the coefficients of the terms  $\mathbb{L}_{B_n}^{-1}$  in the right-hand side of (64), we get

$$\begin{aligned} &\mathbb{L}_{B_n}^{-1} + \sum_{i=2}^n P_i^n (\mathbb{L}_{B_{i-1}}^{-1} - \mathbb{L}_{B_i}^{-1}) - P_1^n \mathbb{L}_{B_1}^{-1} \\ &= (P_2^n - P_1^n) \mathbb{L}_{B_1}^{-1} + (P_3^n - P_2^n) \mathbb{L}_{B_2}^{-1} + \cdots + (P_{n+1}^n - P_n^n) \mathbb{L}_{B_n}^{-1} \\ &= P_2^n \circ \mathbb{L}_{I - \Gamma_1} \circ \mathbb{L}_{B_1}^{-1} + P_3^n \circ \mathbb{L}_{I - \Gamma_2} \circ \mathbb{L}_{B_2}^{-1} + \cdots + P_{n+1}^n \circ \mathbb{L}_{I - \Gamma_n} \circ \mathbb{L}_{B_n}^{-1} \\ &= P_2^n (\gamma_1 \mathbb{I}) + P_3^n (\gamma_2 \mathbb{I}) + \cdots + P_{n+1}^n (\gamma_n \mathbb{I}) \quad [by \text{ Eq. (65)}] \\ &= \sum_{i=1}^n \gamma_i P_{i+1}^n, \end{aligned}$$

where we used the identity  $P_{i+1}^n - P_i^n = P_{i+1}^n \circ (\mathbb{I} - \mathbb{L}_{\Gamma_i}) = P_{i+1}^n \circ \mathbb{L}_{I-\Gamma_i}$ . Furthermore, since  $B_i$  are bounded,

$$\begin{aligned} \|\mathbb{L}_{B_{i-1}}^{-1} - \mathbb{L}_{B_i}^{-1}\| &= \|\mathbb{L}_{B_i}^{-1} \circ (\mathbb{L}_{B_i} - \mathbb{L}_{B_{i-1}}) \circ \mathbb{L}_{B_{i-1}}^{-1}\| = O(\|\mathbb{L}_{B_i} - \mathbb{L}_{B_{i-1}}\|) \\ &= O(\|B_i - B_{i-1}\|) = O(\gamma_i^2). \quad [\text{By assumption of Lemma D.1}] \end{aligned}$$

In addition,  $\|\sum_{i=2}^n P_i^n \circ (\mathbb{L}_{B_{i-1}}^{-1} - \mathbb{L}_{B_i}^{-1})\| \leq K_0 e^{-\Gamma(n)} \sum_{i=2}^n e^{\Gamma(i)} O(\gamma_i^2)$ . Since  $\sum_i O(\gamma_i^2) < \infty$  and  $e^{\Gamma(i)}$  is positive, increasing and diverging, we can invoke Kronecker's lemma and obtain  $\sum_{i=2}^n e^{\Gamma(i)} O(\gamma_i^2) = o(e^{\Gamma(n)})$ . Therefore

$$\sum_{i=2}^n P_i^n \circ (\mathbb{L}_{B_{i-1}}^{-1} - \mathbb{L}_{B_i}^{-1}) \rightarrow \mathbb{L}_0, \quad (66)$$

and since  $P_1^n \rightarrow \mathbb{L}_0$ , we conclude from Equation (65) that

$$\lim_{n \rightarrow \infty} \sum_{i=1}^n \gamma_i P_{i+1}^n = \lim_{n \rightarrow \infty} \mathbb{L}_{B_n}^{-1} = \mathbb{L}_B^{-1}. \quad (67)$$

Thus,  $S_n \rightarrow \mathbb{L}_B^{-1}$ , as desired. For  $\tilde{D}_n$  we have

$$\begin{aligned} \tilde{D}_n &= \sum_{i=1}^n \gamma_i P_{i+1}^n \{D_i\} = \mathbb{L}_{B_n}^{-1} \{D_n\} + \sum_{i=2}^n P_i^n \circ (\mathbb{L}_{B_{i-1}}^{-1} \{D_{i-1}\} - \mathbb{L}_{B_i}^{-1} \{D_i\}) \\ &\quad + P_1^n \circ \mathbb{L}_{B_1}^{-1} \{D_1\}. \end{aligned}$$

Since  $\|D_n\| \rightarrow 0$  it follows that  $\|\mathbb{L}_{B_n}^{-1} \{D_n\}\| \rightarrow 0$  and  $\|(\mathbb{L}_{B_{i-1}}^{-1} \{D_{i-1}\} - \mathbb{L}_{B_i}^{-1} \{D_i\})\| = O(\gamma_i^2)$ . Recall that  $P_1^n \rightarrow \mathbb{L}_0$ , and thus  $\tilde{D}_n \rightarrow \mathbf{0}$ . Finally, we substitute this result in Equation (65) to get  $X_n \rightarrow \mathbb{L}_B^{-1} \{C\}$ .

For the second recursion of the lemma,

$$Y_n = \mathbb{L}_{I+\gamma_n B_n}^{-1} \{Y_{n-1} + \gamma_n (C + D_n)\}, \quad (68)$$

the proof is similar. First, we make the following definitions.

$$\begin{aligned} \Gamma_n &\stackrel{\text{def}}{=} I + \gamma_n B_n, \\ Q_i^n &\stackrel{\text{def}}{=} \mathbb{L}_{\Gamma_n}^{-1} \circ \mathbb{L}_{\Gamma_{n-1}}^{-1} \circ \dots \circ \mathbb{L}_{\Gamma_i}^{-1}. \end{aligned}$$

As before,  $Q_i^n \rightarrow \mathbb{L}_0$ . Solving the recursion (68) yields

$$Y_n = Q_1^n \{Y_0\} + S_n \{C\} + \tilde{D}_n, \quad (69)$$

where we defined  $S_n \stackrel{\text{def}}{=} \sum_{i=1}^n \gamma_i Q_i^n$  and  $\tilde{D}_n \stackrel{\text{def}}{=} \sum_{i=1}^n \gamma_i Q_i^n \{D_i\}$ . The following identities can also be verified by the definition of the linear maps.

$$\mathbb{L}_{B_n}^{-1} \circ (\mathbb{I} - \mathbb{L}_{\Gamma_n}^{-1}) = \gamma_n \mathbb{L}_{\Gamma_n}^{-1}, \quad (70)$$

$$\mathbb{L}_{B_n}^{-1} \mathbb{L}_{\Gamma_n}^{-1} = \mathbb{L}_{\Gamma_n}^{-1} \mathbb{L}_{B_n}^{-1}. \quad (71)$$

It holds,

$$\begin{aligned} \mathbb{L}_{B_n}^{-1} + \sum_{i=1}^n Q_i^n \circ (\mathbb{L}_{B_{i-1}}^{-1} - \mathbb{L}_{B_i}^{-1}) &= \mathbb{L}_{B_n}^{-1} \circ (\mathbb{I} - \mathbb{L}_{\Gamma_n}^{-1}) + \mathbb{L}_{\Gamma_n}^{-1} \circ \mathbb{L}_{B_{n-1}}^{-1} \circ (\mathbb{I} - \mathbb{L}_{\Gamma_n}^{-1}) + \cdots \\ &= \gamma_n \mathbb{L}_{\Gamma_n}^{-1} + \gamma_{n-1} \mathbb{L}_{\Gamma_n}^{-1} \mathbb{L}_{\Gamma_{n-1}}^{-1} + \cdots = S_n, \end{aligned}$$

where the first line is obtained by Eq. (70) and the second line by Eq. (71). Thus, similar to the previously analyzed recursion,  $S_n \rightarrow \mathbb{L}_B^{-1}$  and  $\tilde{D}_n \rightarrow 0$ . Therefore,  $Y_n \rightarrow \mathbb{L}_B^{-1} \{C\}$ .

For both cases, if  $B, C$  commute then  $\mathbb{L}_B^{-1} \{C\} = X$  such that  $BX + XB = 2C$ . Setting  $X = B^{-1}C$  is a solution since  $BB^{-1}C + B^{-1}CB = C + B^{-1}BC = 2C$ . By the Lyapunov theorem, this solution is unique.  $\square$

**Corollary D.1.** *Consider the matrix recursions*

$$X_n = \mathbb{L}_{I - \gamma_n B_n} \{X_{n-1}\} + \gamma_n^2 (C + D_n), \quad (72)$$

$$Y_n = \mathbb{L}_{I + \gamma_n B_n}^{-1} \{Y_{n-1} + \gamma_n^2 (C + D_n)\}, \quad (73)$$

where  $B_n, B, C, D_n$  satisfy the assumptions of Lemma D.1. Moreover, suppose  $\gamma_n = \gamma_1 n^{-1}$ . If the matrix  $B - I/\gamma_1$  is positive-definite, then

$$(1/\gamma_n)X_n, (1/\gamma_n)Y_n \rightarrow \mathbb{L}_{B - I/\gamma_1}^{-1} \{C\} \text{ i.e.,}$$

both matrices  $(1/\gamma_n)X_n$  and  $(1/\gamma_n)Y_n$  approximate the matrix  $\mathbb{L}_{B - I/\gamma_1}^{-1} \{C\}$ . If, in addition,  $B$  and  $C$  commute then  $(1/\gamma_n)X_n, (1/\gamma_n)Y_n \rightarrow (B - I/\gamma_1)^{-1}C$ .

*Proof.* Both  $X_n, Y_n \rightarrow 0$  by direct application of Lemma (D.1). Let  $\tilde{X}_n = (1/\gamma_n)X_n$ . First, divide (72) by  $\gamma_n$  to obtain

$$\tilde{X}_n = \mathbb{L}_{I - \gamma_n B_n} \left\{ \tilde{X}_{n-1} \right\} \frac{\gamma_{n-1}}{\gamma_n} + \gamma_n (C + D_n). \quad (74)$$

By Assumption 3.1(a),  $\gamma_{n-1}/\gamma_n = 1 + \gamma_n/\gamma_1 + O(\gamma_n^2)$ . Then,

$$\mathbb{L}_{I - \gamma_n B_n} \left\{ \tilde{X}_{n-1} \right\} \frac{\gamma_{n-1}}{\gamma_n} = \mathbb{L}_{I - \gamma_n B_n} \left\{ \tilde{X}_{n-1} \right\} + \gamma_n \tilde{X}_{n-1} + O(\gamma_n^2). \quad (75)$$



Therefore, we can rewrite Eq. (74) as

$$\tilde{X}_n = \mathbb{L}_{I - \gamma_n \Gamma_n} \left\{ \tilde{X}_{n-1} \right\} + \gamma_n (C + D_n), \quad (76)$$

where  $\Gamma_n \stackrel{\text{def}}{=} B_n - I/\gamma_1 + O(\gamma_n)$ . In the limit  $\Gamma_n \rightarrow B - I/\gamma_1 > 0$ . Furthermore,  $\|\Gamma_{i-1} - \Gamma_i\| = O(\gamma_i^2)$  by assumptions of Corollary D.1. Thus, we can apply Lemma (D.1) to conclude that  $\tilde{X}_n \stackrel{\text{def}}{=} (1/\gamma_n) X_n \rightarrow \mathbb{L}_{B-I/\gamma_1}^{-1} \{C\}$ . The proof for  $Y_n$  follows the same reasoning since  $(I + \gamma_n B_n)^{-1} (\gamma_{n-1}/\gamma_n) = (I + \gamma_n \Gamma_n)^{-1}$ , where  $\Gamma_n \stackrel{\text{def}}{=} B_n - I/\gamma_1 + O(\gamma_n)$ .  $\square$

**Theorem 3.2.** *Consider SGD procedures (3) and (4), and suppose that Assumptions 3.1(a),(c),(d),(e) hold, where  $\gamma = 1$ . The asymptotic variance of the explicit SGD estimator (3) satisfies*

$$n \text{Var}(\theta_n^{\text{sgd}}) \rightarrow \gamma_1^2 (2\gamma_1 C \mathcal{I}(\theta_\star) - I)^{-1} C \mathcal{I}(\theta_\star) C.$$

*The asymptotic variance of the implicit SGD estimator (4) satisfies*

$$n \text{Var}(\theta_n^{\text{im}}) \rightarrow \gamma_1^2 (2\gamma_1 C \mathcal{I}(\theta_\star) - I)^{-1} C \mathcal{I}(\theta_\star) C.$$

*Proof.* We begin with the implicit SGD procedure. For notational convenience we make the following definitions:  $V_n \stackrel{\text{def}}{=} \text{Var}(\theta_n^{\text{im}})$ ,  $S_n(\theta) \stackrel{\text{def}}{=} \nabla \log f(Y_n; X_n, \theta)$ , Note,  $\mathbb{E}(S_n(\theta)) \stackrel{\text{def}}{=} h(\theta)$ . Let  $J_h$  denote the Jacobian of function  $h$ , then, under typical regularity conditions and by Theorem 3.1:

$$\begin{aligned} \mathbb{E}(S_n(\theta_\star) | X_n) &= 0 \\ \text{Var}(S_n(\theta_\star)) &= \mathbb{E}(\text{Var}(S_n(\theta_\star) | X_n)) \stackrel{\text{def}}{=} \mathcal{I}(\theta_\star) \\ J_h(\theta) &= -\mathcal{I}(\theta), \quad [\text{under regularity conditions}] \\ h(\theta_n^{\text{im}}) &= -\mathcal{I}(\theta_\star)(\theta_n^{\text{im}} - \theta_\star) + O(\gamma_n) \quad [\text{by Theorem 3.1}], \\ \|\text{Var}(S_n(\theta) - S_n(\theta_\star))\| &\leq \mathbb{E}(\|S_n(\theta) - S_n(\theta_\star)\|^2) \leq L_0^2 \mathbb{E}(\|\theta - \theta_\star\|^2). \end{aligned} \quad (77)$$

We can now rewrite Eq. (4) as follows,

$$\theta_n^{\text{im}} = \theta_{n-1}^{\text{im}} + \gamma_n C_n S_n(\theta_n^{\text{im}}) = \theta_{n-1}^{\text{im}} + \gamma_n \lambda_n C_n S_n(\theta_{n-1}^{\text{im}}), \quad (78)$$

where  $\lambda_n$  is defined in Theorem 4.1 and  $\lambda_n = 1 - O(\gamma_n)$  by Eq. (43). Then, taking variances on both sides of Eq. (78) yields

$$V_n = V_{n-1} + \gamma_n^2 C_n \text{Var}(S_n(\theta_n^{\text{im}})) C_n^\top + \gamma_n \text{Cov}(\theta_{n-1}^{\text{im}}, S_n(\theta_n^{\text{im}})) C_n^\top + \gamma_n C_n \text{Cov}(S_n(\theta_n^{\text{im}}), \theta_{n-1}^{\text{im}}). \quad (79)$$

We can simplify all variance/covariance terms in Eq. (79) as follows.

$$\begin{aligned}
C_n \text{Var} \left( S_n(\theta_n^{\text{im}}) \right) C_n^\top &= C_n \text{Var} \left( S_n(\theta_\star) + [S_n(\theta_n^{\text{im}}) - S_n(\theta_\star)] \right) C_n^\top \\
&= C\mathcal{I}(\theta_\star)C^\top + o(1), \quad [\text{by Eqs. (77), Theorem 3.1, and Assumption 3.1(e)}] \\
\text{Cov} \left( \theta_{n-1}^{\text{im}}, S_n(\theta_n^{\text{im}}) \right) &= \text{Cov} \left( \theta_{n-1}^{\text{im}}, S_n(\theta_{n-1}^{\text{im}}) \right) + \text{Cov} \left( \theta_{n-1}^{\text{im}}, (\lambda_n - 1)S_n(\theta_{n-1}^{\text{im}}) \right) \\
&= \text{Cov} \left( \theta_{n-1}^{\text{im}}, h(\theta_{n-1}^{\text{im}}) \right) + O(\gamma_n) \\
&= V_{n-1}\mathcal{I}(\theta_\star) + O(\gamma_n). \quad [\text{by Eq. (77), Theorem 3.1, Eq. (43)}].
\end{aligned}$$

Similarly,  $\text{Cov} \left( h(\theta_n^{\text{im}}), \theta_{n-1}^{\text{im}} \right) = V_{n-1}\mathcal{I}(\theta_\star) + O(\gamma_n)$ . We can now rewrite Eq. (79) as

$$V_n = \mathbb{L}_{I - \gamma_n B_n} \{V_{n-1}\} + \gamma_n^2 [C\mathcal{I}(\theta_\star)C^\top + o(1)], \quad (80)$$

where  $B_n \stackrel{\text{def}}{=} 2C_n\mathcal{I}(\theta_\star)$  and  $B_n \rightarrow 2C\mathcal{I}(\theta_\star)$ . Corollary (D.1) on recursion (80) yields

$$(1/n)V_n \rightarrow \gamma_1^2 (2\gamma_1 C\mathcal{I}(\theta_\star) - I)^{-1} C\mathcal{I}(\theta_\star)C.$$

We obtain the asymptotic variance in closed-form since  $B$  and  $C$  commute, and  $C$  is symmetric. The regularity conditions (77) and the convergence rates of Theorem 3.1 that are crucial for this proof, also hold for the explicit procedure. Therefore for the proof for the asymptotic variance of explicit SGD (3) is identical.  $\square$

**Theorem 3.3.** *Consider SGD procedure (13) and suppose Assumptions 3.1(a),(c),(d), and (e) hold, where  $\gamma \in [0.5, 1)$ . Then, the iterate  $\theta_n^{\text{im}}$  converges to  $\theta_\star$  in probability and is asymptotically efficient, i.e.,*

$$n\text{Var} \left( \overline{\theta_n^{\text{im}}} \right) \rightarrow \mathcal{I}(\theta_\star)^{-1}.$$

*Proof.* By Theorem 3.1 and Assumptions 3.1 (d), (c), we have

$$\nabla \log f(Y_n; X_n, \theta_n^{\text{im}}) = \nabla \log f(Y_n; X_n, \theta_\star) - \mathcal{I}(\theta_\star)(\theta_n^{\text{im}} - \theta_\star) + O(\gamma_n). \quad (81)$$

Define, for convenience  $\varepsilon_n \stackrel{\text{def}}{=} \nabla \log f(Y_n; X_n, \theta_\star)$ ,  $F \stackrel{\text{def}}{=} \mathcal{I}(\theta_\star)$ . Then, the first-order implicit SGD iteration becomes

$$\theta_n^{\text{im}} - \theta_\star = (I + \gamma_n F)^{-1}(\theta_{n-1}^{\text{im}} - \theta_\star + \gamma_n \varepsilon_n + O(\gamma_n^2)). \quad (82)$$

We make the following definitions.

$$\begin{aligned}
e_i &\stackrel{\text{def}}{=} \gamma_i (I + \gamma_i F)^{-1} (\varepsilon_i + O(\gamma_i^2)), \\
B_i^j &\stackrel{\text{def}}{=} \prod_{k=j}^i (I + \gamma_k F)^{-1}, \\
D_j^n &\stackrel{\text{def}}{=} \prod_{k=n-1}^i B_{j+1}^k = I + B_{j+1}^{j+1} + B_{j+1}^{j+2} + \dots + B_{j+1}^{n-1}.
\end{aligned} \tag{83}$$

Then, we can solve the recursion for  $\overline{\theta_n^{\text{im}}} - \theta_*$  to obtain

$$\overline{\theta_n^{\text{im}}} - \theta_* = (1/n) D_0^n (\overline{\theta_n^{\text{im}}} - \theta_*) + (1/n) \sum_i^{n-1} D_i^n e_i. \tag{84}$$

Our proof is now split into proving the following two lemmas.

**Lemma D.2.** *Under Assumption 3.1(a)  $D_0^n = o(n)$ .*

*Proof.* Matrix  $F$  is positive-definite by Assumption 3.1(d). Thus, if  $\lambda$  is some eigenvalue of  $F$  then the corresponding eigenvalue of  $D_0^n$  is  $1 + \frac{1}{1+\gamma_1\lambda} + \frac{1}{1+\gamma_1\lambda} \frac{1}{1+\gamma_2\lambda} + \dots \leq \sum_{i=0}^n \exp(-K\lambda \sum_{k=1}^i \gamma_k)$ , where the last inequality is obtained by Lemma B.1. Because  $\sum \gamma_i \rightarrow \infty$ , the summands are  $o(1)$ , and thus  $D_0^n$  is  $o(n)$ .  $\square$

**Lemma D.3.** *Suppose Assumption 3.1(a) and Eq. (81) hold. Then,*

$$\gamma_i D_i^n (I + \gamma_i F)^{-1} = \Omega_i^n + F^{-1}, \tag{85}$$

such that  $\sum_{i=0}^{n-1} \Omega_i^n = o(n)$ .

*Proof.* Our goal will be to compare the eigenvalues of  $\gamma_i D_i^n$  and  $F$ . Any matrix  $D_i^n$  shares the same eigenvectors with  $F$  because  $F$  is positive-definite, and thus a relationship on eigenvalues will automatically establish a relationship on the matrices. For convenience, define  $q_i^j \stackrel{\text{def}}{=} \prod_{k=i}^j (1 + \gamma_k \lambda)^{-1}$  for  $\lambda > 0$ ; by convention,  $q_{i-1}^i = 1$ . Also let  $s_i^j \stackrel{\text{def}}{=} \sum_{k=i}^j \gamma_k$  be the function of partial sums. By Lemma B.1  $q_i^j = O(\exp(-K\lambda s_i^j))$ , for some  $K > 0$ . For an eigenvalue  $\lambda > 0$  of  $F$  the corresponding eigenvalue, say  $\lambda'$ , of matrix  $\gamma_i D_i^n (I + \gamma_i F)^{-1}$  is equal to

$$\lambda' = \frac{\gamma_i}{1 + \gamma_i \lambda} (q_{i+1}^i + q_{i+1}^{i+1} + \dots + q_{i+1}^{n-1}). \tag{86}$$

Thus,

$$\lambda' (1 + \gamma_i \lambda) = \sum_{k=i}^{n-1} \gamma_i q_{i+1}^k. \tag{87}$$

Our goal will be to derive the relationship between  $\lambda$  and  $\lambda'$ . By definition

$$\begin{aligned}
\gamma_{i+1}\lambda q_{i+1}^{i+1} + q_{i+1}^{i+1} &= 1 \\
\gamma_{i+2}\lambda q_{i+1}^{i+2} + q_{i+1}^{i+2} &= q_{i+1}^{i+1} \\
&\dots\dots\dots \\
\gamma_{n-2}\lambda q_{i+1}^{n-2} + q_{i+1}^{n-2} &= q_{i+1}^{n-3} \\
\gamma_{n-1}\lambda q_{i+1}^{n-1} + q_{i+1}^{n-1} &= q_{i+1}^{n-2}.
\end{aligned} \tag{88}$$

By summing over the terms we obtain:

$$\lambda \sum_{k=i+1}^{n-1} \gamma_k q_{i+1}^k + q_{i+1}^{n-1} = 1. \tag{89}$$

If we combine with (86) we obtain

$$\lambda \sum_{k=i}^{n-1} \gamma_i q_{i+1}^k + \lambda \sum_{k=i}^{n-1} (\gamma_k - \gamma_i) q_{i+1}^k + q_{i+1}^{n-1} = 1 + \gamma_i \lambda \Rightarrow \tag{90}$$

$$(1 + \gamma_i \lambda) \lambda \lambda' + \lambda \sum_{k=i}^{n-1} (\gamma_k - \gamma_i) q_{i+1}^k + q_{i+1}^{n-1} = 1 + \gamma_i \lambda. \tag{91}$$

We now focus on the second term. By telescoping the series we obtain

$$\begin{aligned}
\lambda \sum_{k=i}^{n-1} (\gamma_k - \gamma_i) q_{i+1}^k &= \lambda \sum_{k=i}^{n-1} \left[ \sum_{j=i}^k (\gamma_{j+1} - \gamma_j) \right] q_{i+1}^k = \lambda \sum_{k=i}^{n-1} \left[ \sum_{j=i}^k \gamma_j o(\gamma_j) \right] q_{i+1}^k \\
&\leq \lambda o(\gamma_i) \sum_{k=i}^{n-1} s_i^k q_{i+1}^k \triangleq q_i^n.
\end{aligned} \tag{92}$$

In Eq. (92) we used  $(\gamma_{j+1} - \gamma_j)/\gamma_j = O(n^{-1-\gamma})/n^{-\gamma} = O(n^{-1}) = o(\gamma_j)$ , by Assumption 3.1(a). Our goal is now to show  $\sum_{i=0}^{n-1} q_i^n = o(n)$ . Since  $q_{i+1}^k = O(\exp(-K\lambda s_{i+1}^k))$  by Polyak (1992, p845, see A6 and A7) we obtain that  $q_i^n \rightarrow 0$  for fixed  $i$  as  $n \rightarrow \infty$ . Therefore we can rewrite (90) as

$$\lambda' \lambda + q_i^n + O(q_{i+1}^n) = 1, \tag{93}$$

where  $\sum_{i=0}^n q_{i+1}^n = o(n)$  and  $\sum_{i=0}^{n-1} q_i^n = o(n)$ .  $\square$

Our proof is now complete. By Eq. (84) and Lemmas D.2 and D.3 we have

$$\overline{\theta}_n^{\text{im}} - \theta_\star = F^{-1} \sum_{i=1}^n \varepsilon_i + (1/n) o(n).$$

Because  $\text{Var}(\varepsilon_i) = \mathcal{I}(\theta_*)$ , we finally obtain

$$n\text{Var}\left(\overline{\theta_n^{\text{im}}} - \theta_*\right) = \mathcal{I}(\theta_*)^{-1}.$$

□

**Theorem 3.4.** *Suppose that Assumptions 3.1(a),(c),(d),(e),(f) hold. Then, the iterate  $\theta_n^{\text{im}}$  of implicit SGD (4) is asymptotically normal, such that*

$$n^{\gamma/2}(\theta_n^{\text{im}} - \theta_*) \rightarrow \mathcal{N}_p(0, \Sigma),$$

where  $\Sigma = \gamma_1^2 (2\gamma_1 C\mathcal{I}(\theta_*) - I)^{-1} C\mathcal{I}(\theta_*)C$ .

*Proof.* Let  $S_n(\theta) \stackrel{\text{def}}{=} \nabla \log f(Y_n; X_n, \theta)$  as in the proof of Theorem 3.2. The conditions for Fabian's theorem – see Fabian (1968b, Theorem 1) – hold also for the implicit procedure. The goal is to show that

$$\theta_n^{\text{im}} - \theta_* = (I - \gamma_n A_n)(\theta_{n-1}^{\text{im}} - \theta_*) + \gamma_n \xi_n(\theta_*) + O(\gamma_n^2), \quad (94)$$

where  $A_n \rightarrow A \succeq 0$ , and  $\xi_n(\theta) = S_n(\theta) - h(\theta)$ , and  $h(\theta) = \mathbb{E}(S_n(\theta))$ ; note,  $\xi_n(\theta_*) = S_n(\theta_*)$ . Indeed, by a Taylor expansion on  $S_n(\theta_n^{\text{im}})$  and considering that  $\theta_n^{\text{im}} = \theta_{n-1}^{\text{im}} + \gamma_n S_n(\theta_n^{\text{im}})$ , by definition, we have

$$(I + \gamma_n \hat{\mathcal{I}}_n(\theta_*))(\theta_n^{\text{im}} - \theta_*) = \theta_{n-1}^{\text{im}} - \theta_* + \gamma_n S_n(\theta_*), \quad (95)$$

where  $\hat{\mathcal{I}}_n(\theta_*) = -\nabla^2 S_n(\theta_*)$ ; note,  $\mathbb{E}(\hat{\mathcal{I}}_n(\theta_*)) = \mathcal{I}(\theta_*)$ . Because  $(I + \gamma_n \hat{\mathcal{I}}_n(\theta_*))^{-1} = I - \gamma_n \hat{\mathcal{I}}_n(\theta_*) + O(\gamma_n^2)$ , we can rewrite Eq. (95) as

$$\theta_n^{\text{im}} - \theta_* = (I - \gamma_n \hat{\mathcal{I}}_n(\theta_*))(\theta_{n-1}^{\text{im}} - \theta_*) + \gamma_n S_n(\theta_*) + O(\gamma_n^2). \quad (96)$$

We can now apply Fabian's Theorem to derive asymptotic normality of  $\theta_n^{\text{im}}$ . The variance matrix of the asymptotic normal distribution is derived in Theorem 3.4 under weaker conditions. □

## E Stability

**Theorem 3.1.** *Let  $\bar{\lambda}_f = \max \text{eig}(\mathcal{I}(\theta_*))$ , and suppose  $\gamma_n = \gamma_1/n$  and  $\gamma_1 \bar{\lambda}_f > 1$ . Then, the maximum eigenvalue of  $P_1^n$  satisfies*

$$\max_{n>0} \max \{\text{eig}(P_1^n)\} = \Theta(2^{\gamma_1 \bar{\lambda}_f} / \sqrt{\gamma_1 \bar{\lambda}_f}).$$

For the implicit method,

$$\max_{n>0} \max \{\text{eig}(Q_1^n)\} = O(1).$$

*Proof.* We will use the following intermediate result:

$$\max_{n>0} \left| \prod_{i=1}^n (1 - b/i) \right| \approx \begin{cases} 1 - b & \text{if } 0 < b < 1 \\ \frac{2^b}{\sqrt{2\pi b}} & \text{if } b > 1 \end{cases}$$

The first case is obvious. For the second case,  $b > 1$ , assume without loss of generality that  $b$  is an even integer. Then the maximum is given by

$$(b-1)(b/2-1)(b/3-1)\cdots(2-1) = \frac{1}{2} \binom{b}{b/2} = \Theta(2^b/\sqrt{2\pi b}), \quad (97)$$

where the last approximation follows from Stirling's formula. The stability result on the explicit SGD updates of Lemma 3.1 follows immediately by using the largest eigenvalue  $\overline{\lambda}_f$  of  $\mathcal{I}(\theta_*)$ . For the implicit SGD updates, we note that the eigenvalues of  $(I + \gamma_n \mathcal{I}(\theta_*))^{-1}$  are less than one, for any  $\gamma_n > 0$  and any Fisher matrix.  $\square$

## F Applications

**Theorem (4.1).** *Suppose Assumption 3.1(b) holds. Then, the gradient for the implicit update (4) is a scaled version of the gradient at the previous iterate, i.e.,*

$$\nabla \log f(Y_n; X_n, \theta_n^{\text{im}}) = \lambda_n \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{im}}), \quad (98)$$

where the scalar  $\lambda_n$  satisfies,

$$\lambda_n \ell'(X_n^\top \theta_{n-1}^{\text{im}}; Y_n) = \ell'(X_n^\top \theta_{n-1}^{\text{im}} + \gamma_n \lambda_n \ell'(X_n^\top \theta_{n-1}^{\text{im}}; Y_n) X_n^\top C_n X_n; Y_n). \quad (99)$$

*Proof.* From the chain rule  $\nabla \log f(Y_n; X_n, \theta) = \ell'(X_n^\top \theta; Y_n) X_n$ , and thus  $\nabla \log f(Y_n; X_n, \theta_n^{\text{im}}) = \ell'(X_n^\top \theta_n^{\text{im}}; Y_n) X_n$  and  $\nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{im}}) = \ell'(X_n^\top \theta_{n-1}^{\text{im}}; Y_n) X_n$ , and thus the two gradients are colinear. Therefore there exists a scalar  $\lambda_n$  such that

$$\begin{aligned} \nabla \log f(Y_n; X_n, \theta_n^{\text{im}}) &= \lambda_n \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{im}}) \Rightarrow \\ \ell'(X_n^\top \theta_n^{\text{im}}; Y_n) X_n &= \lambda_n \ell'(X_n^\top \theta_{n-1}^{\text{im}}; Y_n) X_n. \end{aligned} \quad (100)$$

We also have,

$$\begin{aligned} \theta_n^{\text{im}} &= \theta_{n-1}^{\text{im}} + \gamma_n C_n \log f(Y_n; X_n, \theta_n^{\text{im}}) \quad [\text{by definition of implicit SGD (4)}] \\ &= \theta_{n-1}^{\text{im}} + \gamma_n \lambda_n C_n \log f(Y_n; X_n, \theta_{n-1}^{\text{im}}). \quad [\text{by Eq. (100)}] \end{aligned} \quad (101)$$

Substituting the expression for  $\theta_n^{\text{im}}$  in Eq.(101) into Eq. (100) we obtain the desired result of the Theorem in Eq. (98).

We now prove the last claim of the theorem regarding the search bounds for  $\lambda_n$ . For notational convenience, define  $a \stackrel{\text{def}}{=} X_n^\top \theta_{n-1}^{\text{im}}$ ,  $g(x) \stackrel{\text{def}}{=} \ell'(x; Y_n)$ , and  $c = X_n^\top C_n X_n$ , where  $c > 0$  because  $C_n$  are positive-definite. Also let  $x_\star \stackrel{\text{def}}{=} \gamma_n \lambda_n g(a)$ , then the fixed-point equation (99) can be written as

$$x_\star = \gamma_n g(a + x_\star c). \quad (102)$$

where  $g$  is decreasing by Assumption (b). If  $g(a) = 0$  then  $x_\star = 0$ . If  $g(a) > 0$  then  $x_\star > 0$  and  $\gamma_n g(a + xc) < \gamma_n g(a)$  for all  $x > 0$ , since  $g(a + xc)$  is decreasing; taking  $x = x_\star$  yields  $\gamma_n g(a) > \gamma_n g(a + x_\star c) = x_\star$ , by the fixed-point equation (102). Thus,  $0 < x_\star < \gamma_n g(a)$ . Similarly, if  $g(a) < 0$  then  $x_\star < 0$  and  $\gamma_n g(a + xc) > \gamma_n g(a)$  for all  $x < 0$ , since  $g(a + xc)$  is decreasing; taking  $x = x_\star$  yields  $\gamma_n g(a) < \gamma_n g(a + x_\star c) = x_\star$ , by the fixed-point equation. Thus,  $\gamma_n g(a) < x_\star < 0$ . In both cases  $0 < \lambda_n < 1$ . A visual proof is given Figure 6.

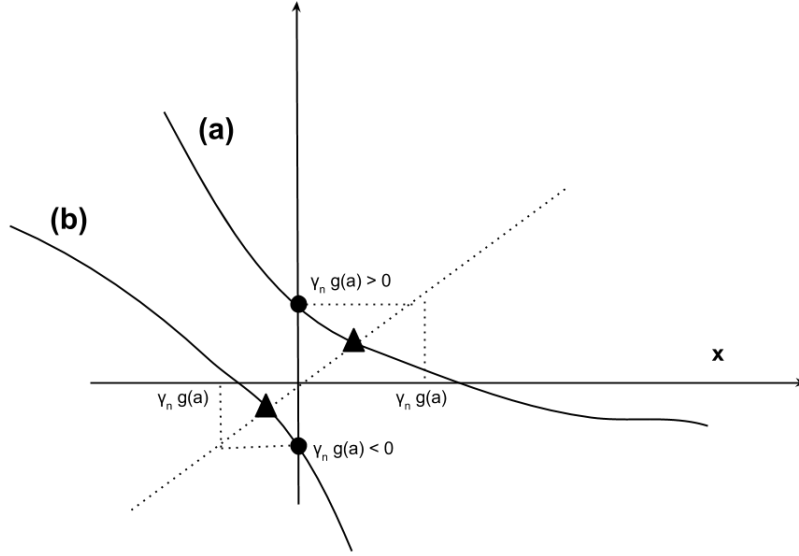


Figure 6: (Search bounds for solution of Eq. (102)) **Case  $g(a) > 0$ :** Corresponds to Curve (a) defined as  $\gamma_n g(a + xc)$ ,  $c > 0$ . The solution  $x_\star$  of fixed point equation (102) (corresponding to right triangle) is between 0 and  $\gamma_n g(a)$  since Curve (a) is decreasing. **Case  $g(a) < 0$ :** Corresponds to Curve (b) also defined as  $\gamma_n g(a + xc)$ . The solution  $x_\star$  of fixed point equation (102) (left triangle) is between  $\gamma_n g(a)$  and 0 since Curve (b) is also decreasing.

□

## G Additional experiments

### G.1 Normality experiments with implicit SGD

In Figure 7 we plot the experimental results of Section 5.1.2 for  $p = 50$  (parameter dimension). We see that explicit SGD becomes even more unstable in more dimensions as expected. In contrast, implicit SGD remains stable and validates the theoretical normal distribution for small learning rates. In larger learning rates we observe a divergence from the asymptotic chi-squared distribution (e.g.,  $\gamma_1 = 6$ ) because when the learning rate parameter is large there is more noise in the stochastic approximations, and thus more iterations are required for convergence. In this experiment we fixed the number of iterations for each value of the learning rate, but subsequent experiments verified that implicit SGD reaches the theoretical chi-squared distribution if the number of iterations is increased. Finally, in Figure 8 we make a similar plot for a logistic regression model. In this case the learning rates need to be larger because with the same distribution of covariates for  $X_n$ , the Fisher information is smaller than in the linear normal model. Thus, in almost all experiments explicit SGD was unstable and could not converge whereas implicit SGD was stable and followed the theoretical chi-squared distribution.

### G.2 Poisson regression

In this experiment, we illustrate our method on a bivariate Poisson model which is simple enough to derive the variance formula analytically. We assume binary features such that, for any iteration  $n$ ,  $X_n$  is either  $(0, 0)^\top$ ,  $(1, 0)^\top$  or  $(0, 1)^\top$  with probabilities 0.6, 0.2 and 0.2 respectively. We set  $\theta_\star = (\theta_1, \theta_2)^\top$  for some  $\theta_1, \theta_2$ , and assume  $Y_n \sim \text{Poisson}(\exp(X_n^\top \theta_\star))$ , where the transfer function  $h$  is the exponential, i.e.,  $h(x) = \exp(x)$ . It follows,

$$\mathcal{I}(\theta_\star) = \mathbb{E}(h'(X_n^\top \theta_\star) X_n X_n^\top) = 0.2 \begin{pmatrix} e^{\theta_1} & 0 \\ 0 & e^{\theta_2} \end{pmatrix}.$$

We set  $\gamma_n = 10/3n$  and  $C_n = I$ . Setting  $\theta_1 = \log 2$  and  $\theta_2 = \log 4$ , the asymptotic variance  $\Sigma$  in Theorem 3.2 is equal to

$$\Sigma = \frac{2}{3} \begin{pmatrix} \frac{e^{\theta_1}}{(4/3)e^{\theta_1}-1} & 0 \\ 0 & \frac{e^{\theta_2}}{(4/3)e^{\theta_2}-1} \end{pmatrix} = \begin{pmatrix} 0.8 & 0 \\ 0 & 0.62 \end{pmatrix}. \quad (103)$$

Next, we obtain 100 independent samples of  $\theta_n^{\text{sgd}}$  and  $\theta_n^{\text{im}}$  for  $n = 20000$  iterations of procedures (3) and (4), and compute their empirical variances. We observe



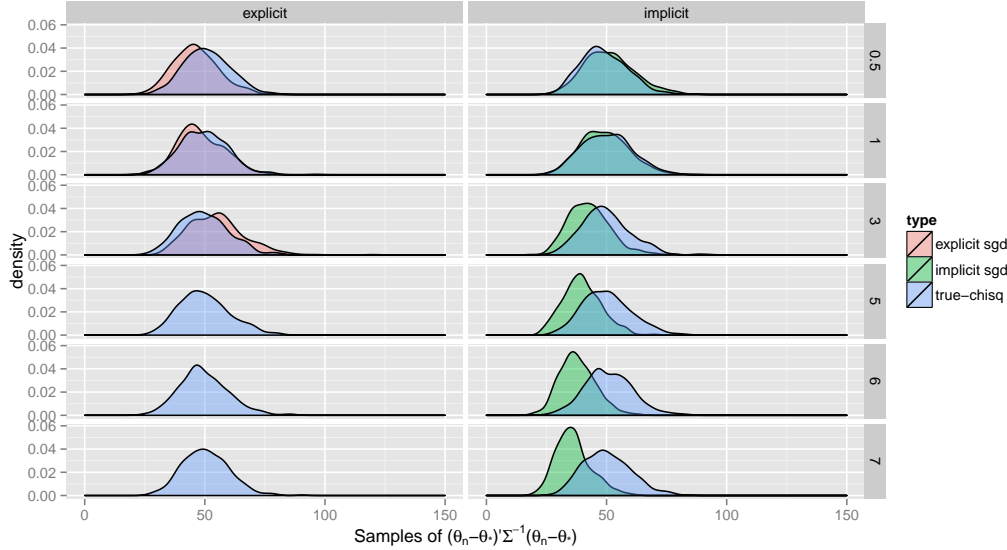


Figure 7: Simulation with normal model for  $p = 50$  parameters. Implicit SGD is stable and follows the nominal chi-squared distribution well, regardless of the particular learning rate. Explicit SGD becomes unstable at larger  $\gamma_1$  and its distribution does not follow the theoretical distribution chi-squared distribution well. In particular, the distribution of  $N(\theta_N^{\text{sgd}} - \theta_*)^\top \Sigma^{-1} (\theta_N^{\text{sgd}} - \theta_*)$  quickly becomes unstable for larger values of the learning rate parameter, and eventually diverges when  $\gamma_1 > 3$ .

that the implicit estimates are particularly stable and have an empirical variance satisfying

$$(1/\gamma_n) \widehat{\text{Var}}(\theta_n^{\text{im}}) = \begin{pmatrix} 0.86 & -0.06 \\ -0.06 & 0.64 \end{pmatrix},$$

and that is close to the theoretical value (103). In contrast, the standard SGD estimates are quite unstable and their  $L_2$  distance to the true values  $\theta_*$  are orders of magnitude larger than the implicit ones (see Table 4 for sample quantiles). By Lemma 3.1, such deviations are expected for standard SGD because the largest eigenvalue of  $\mathcal{I}(\theta_*)$  is  $\lambda_{(2)} = 0.8$  satisfying  $\gamma_1 \lambda_{(2)} = 8/3 > 1$ . Note that it is fairly straightforward to stabilize the standard SGD procedure in this problem, for example by modifying the learning rate sequence to  $\gamma_n = \min\{0.15, 10/3n\}$ . In general, when the optimization problem is well-understood, it is easy to determine the learning rate schedule that avoids out-of-bound explicit updates. In practice, however, we are working with problems that are not so well-understood and determining the correct learning rate parameters may take substantial effort, especially in multi-dimensional settings. The implicit method eliminates this

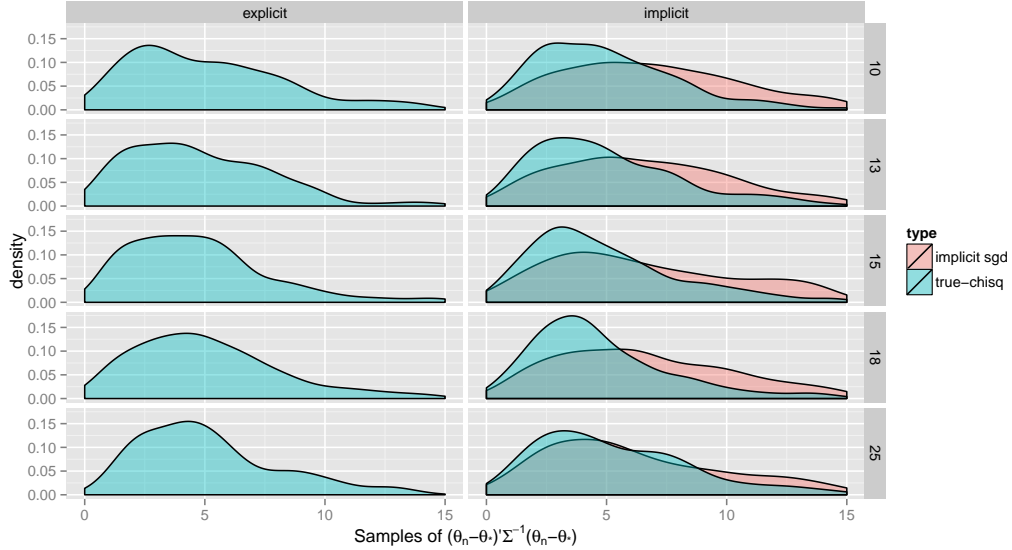


Figure 8: Simulation with logistic regression model for  $p = 5$ . Learning rates are larger than in the linear normal model to ensure the asymptotic covariance matrix of Theorem 3.2 is positive definite. Implicit SGD is stable and follows the nominal chi-squared distribution regardless of the learning rate. Explicit SGD is unstable at virtually all replications of this experiment.

overhead; a wide range of learning rates can lead to convergence on all problems.

### G.3 Experiments with glmnet

In this section, we transform the outcomes in the original experiment  $Y$  through the logistic transformation and then fit a logistic regression model. The results are shown in Table 5 which replicates and expands on Table 2 of (Friedman et al., 2010). The implicit SGD method maintains a stable running time over different correlations and scales sub-linearly in the model size  $p$ . In contrast, `glmnet` is affected by the model size  $p$  and covariate correlation such that it remains 2x-10x slower across experiments. The SGD method is significantly slower in the logistic regression example compared to the normal case (Table 5). This is because the implicit equation of Algorithm 1 needs to be solved numerically, whereas a closed-form solution is available in the normal case.

### G.4 Support vector machines

In this experiment, we are interested to test the performance of the implicit procedure outside the family of GLMs. For that purpose, we implement an implicit

Table 4: Quantiles of  $\|\theta_n^{\text{sgd}} - \theta_\star\|$  and  $\|\theta_n^{\text{im}} - \theta_\star\|$ . Values larger than  $1\text{e}3$  are marked “\*”.

METHOD	QUANTILES					
	25%	50%	75%	85%	95%	100%
SGD	0.01	1.3	435.8	*	*	*
IMPLICIT	0.00	0.01	0.02	0.02	0.03	0.04

online learning procedure for a SVM model and compare it to a standard SGD method on the RCV1 dataset, which is a typical large-scale machine learning benchmark.<sup>13</sup> Some results using variations on the loss functions and the regularization parameter are shown in Table 6. A complete understanding of these results is still missing, however we do observe that the implicit method fares well compared to optimized explicit SGD and, at the same time, remains remarkably robust to misspecification. For example, in all experiments the standard SGD method degrades in performance for small or large regularization (in these experiments, the regularization parameter  $\lambda$  also affects the learning rate such that larger  $\lambda$  means larger learning rates). However, the implicit method maintains a more stable performance accross experiments and, interestingly, it achieves best performance under minimal regularization using the hinge loss.

## G.5 Experiments with machine learning algorithms

In this section we perform additional experiments with related methods from the machine learning literature. We focus on averaged implicit SGD (13), which was shown to achieve optimality under suitable conditions, because most machine learning methods are also designed to achieve optimality (also known linear convergence rate), in the context of maximum-likelihood (or maximum a-posteriori) computation with a finite data set. In summary, our experiments include the following procedures:

- Explicit SGD procedure (3).
- Implicit SGD procedure (4).
- Averaged explicit SGD: Averaged stochastic gradient descent with explicit updates of the iterates (Xu, 2011; Shamir and Zhang, 2012; Bach and

---

<sup>13</sup> We used Bottou’s SVM SGD implementation available at <http://leon.bottou.org/projects/sgd>. Our implicit SVM is available at the first author’s website.

Moulines, 2013). This is equivalent to procedure (13) where the implicit update is replaced by an explicit one,  $\theta_n^{\text{sgd}} = \theta_{n-1}^{\text{sgd}} + \gamma_n \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{sgd}})$ .

- **Prox-SVRG**: A proximal version of the stochastic gradient descent with progressive variance reduction (SVRG) method (Xiao and Zhang, 2014).
- **Prox-SAG**: A proximal version of the stochastic average gradient (SAG) method (Schmidt et al., 2013). While its theory has not been formally established, **Prox-SAG** has shown similar convergence properties to **Prox-SVRG**.<sup>14</sup>
- **AdaGrad** (Duchi et al., 2011b) as defined in Eq. (11). We note that **AdaGrad** and similar adaptive methods effectively approximate the natural gradient by using a larger-dimensional learning rate. It has the added advantage of being less sensitive than first-order methods to tuning of hyperparameters.

### G.5.1 Averaged explicit SGD

In this experiment we validate the theory of statistical efficiency and stability of averaged implicit SGD. To do so, we follow a simple normal linear regression example from Bach and Moulines (2013), which is similar to the experiment in Section 5.2.1. We set  $N = 1\text{e}6$  as the number of observations, and  $p = 20$  be the number of covariates. We also set  $\theta_\star = (0, 0, \dots, 0)^\top \in \mathbb{R}^{20}$  as the true parameter value. The random variables  $X_n$  are distributed i.i.d. as  $X_n \sim \mathcal{N}_p(0, H)$ , where  $H$  is a randomly generated symmetric matrix with eigenvalues  $1/k$ , for  $k = 1, \dots, p$ . The outcome  $Y_n$  is sampled from a normal distribution as  $Y_n | X_n \sim \mathcal{N}(X_n^\top \theta_\star, 1)$ , for  $n = 1, \dots, N$ . We choose a constant learning rate  $\gamma_n \equiv \gamma$  according to the average radius of the data  $R^2 = \text{trace}(H)$ , and for both averaged explicit and implicit SGD we collect iterates  $\theta_n$  for  $n = 1, \dots, N$ , and keep the average  $\bar{\theta}_n$ . In Figure 9, we plot  $(\theta_n - \theta_\star)^\top H(\theta_n - \theta_\star)$  for each iteration for a maximum of  $N$  iterations, i.e., a full pass over the data, in log-log space.

Figure 9 shows that averaged implicit SGD performs on par with averaged explicit SGD for the rates at which averaged explicit SGD is known to be optimal. Thus, averaged implicit SGD is also optimal. However, the benefit of the implicit procedure in averaged implicit SGD becomes clear as the learning rate deviates; notably, averaged implicit SGD remains stable for learning rates that are above the theoretical threshold, i.e.,  $\gamma > 1/R^2$ , whereas averaged explicit SGD diverges in the case of  $\gamma = 2/R^2$ . This stable behavior is also exhibited in implicit SGD, but it converges at a slower rate than averaged implicit SGD, and thus cannot effectively combine stability with statistical efficiency. We note that stability of

<sup>14</sup>We note that the linear convergence rates for **Prox-SVRG** and **Prox-SAG** refer to convergence to the empirical minimizer (i.e., MLE or MAP), and not to the ground truth  $\theta_\star$ .

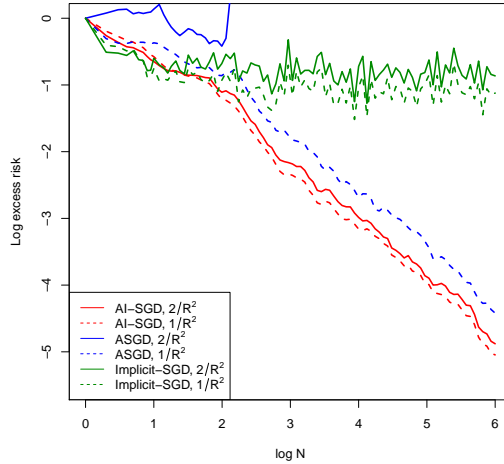


Figure 9: Loss of averaged implicit SGD, averaged explicit SGD, and plain implicit SGD (4) ( $C_n = I$ ), on simulated multivariate normal data with  $N = 1\text{e}6$  observations  $p = 20$  features. The plot shows that averaged implicit SGD is stable regardless of the specification of the learning rate  $\gamma$  and without sacrificing performance. In contrast, explicit averaged SGD is very sensitive to misspecification of the learning rate.

averaged implicit SGD is also observed in the same experiments using decaying learning rates.

### G.5.2 Prox-SVRG and Prox-SAG

We now conduct a study of averaged implicit SGD’s empirical performance on the standard benchmark of large scale linear classification on real data sets. For brevity, we display results on four data sets, although we have seen similar results on eight additional ones.

Our datasets are summarized in Table 7. The COVTYPE data set (Blackard, 1998) consists of forest cover types in which the task is to classify class 2 among 7 forest cover types. DELTA is synthetic data offered in the PASCAL Large Scale Challenge (Sonnenburg et al., 2008) and we apply the default processing offered by the challenge organizers. The task in RCV1 is to classify documents belonging to class CCAT in the text dataset (Lewis et al., 2004), where we apply the standard preprocessing provided by Bottou (2012). In the MNIST data set

(Le Cun et al., 1998) of images of handwritten digits, the task is to classify digit 9 against all others.

For averaged implicit SGD and averaged explicit SGD, we use the learning rate  $\gamma_n = \eta_0(1 + \lambda\eta_0n)^{-3/4}$  prescribed in Xu (2011), where the constant  $\eta_0$  is determined using a small subset of the data. Hyperparameters for other methods are set based on a computationally intensive grid search over the entire hyperparameter space: for **Prox-SVRG**, this includes the step size  $\eta$  in the proximal update and the inner iteration count  $m$ , and for **Prox-SAG**, the same step size  $\eta$ .

The results are shown in Figure 10. We see that averaged implicit SGD achieves comparable performance with the tuned proximal methods **Prox-SVRG** and **Prox-SAG**, as well as **AdaGrad**. All methods have a comparable convergence rate and take roughly a single pass in order to converge. **AdaGrad** exhibits a larger variance in its estimate than the proximal methods, which can be explained from our theoretical results in Section 3.2.1. We also note that as averaged implicit SGD achieves comparable results to the other proximal methods, it also requires no tuning while **Prox-SVRG** and **Prox-SAG** do require careful tuning of their hyperparameters. This was confirmed from separate sensitivity analyses (not reported in this paper), which indicated that aisgd is robust to fine-tuning of hyperparameters in the learning rate, whereas small perturbations of hyperparameters in averaged explicit SGD (the learning rate), **Prox-SVRG** (proximal step size  $\eta$  and inner iteration  $m$ ), and **Prox-SAG** (proximal step size  $\eta$ ), can lead to arbitrarily bad error rates.

Table 5: Experiments comparing implicit SGD with `glmnet`. Covariates  $X$  are sampled as normal, with cross-correlation  $\rho$ , and the outcomes are sampled as  $\mathbf{y} \sim \text{Binom}(\mathbf{p})$ ,  $\text{logit}(\mathbf{p}) = \mathcal{N}(X\theta_*, \sigma^2 I)$ . Running times (in secs) are reported for different values of  $\rho$  averaged over 10 repetitions.

METHOD	METRIC	CORRELATION ( $\rho$ )			
		0	0.2	0.6	0.9
<hr/>					
$N = 1000, p = 10$					
<hr/>					
GLMNET	TIME(SECS)	0.02	0.02	0.026	0.051
	MSE	0.256	0.257	0.292	0.358
SGD	TIME(SECS)	0.058	0.058	0.059	0.062
	MSE	0.214	0.215	0.237	0.27
<hr/>					
$N = 5000, p = 50$					
<hr/>					
GLMNET		0.182	0.193	0.279	0.579
		0.131	0.139	0.152	0.196
SGD		0.289	0.289	0.296	0.31
		0.109	0.108	0.116	0.14
<hr/>					
$N = 100000, p = 200$					
<hr/>					
GLMNET		8.129	8.524	9.921	22.042
		0.06	0.061	0.07	0.099
SGD		5.455	5.458	5.437	5.481
		0.045	0.046	0.048	0.058

Table 6: Test errors of standard and implicit SGD methods on the RCV1 dataset benchmark. Training times are roughly comparable. Best scores, for a particular loss and regularization, are in bold.

		REGULARIZATION ( $\lambda$ )		
LOSS		1E-5	1E-7	1E-12
HINGE	SGD	<b>4.65%</b>	<b>3.57%</b>	4.85%
	IMPLICIT	4.68%	3.6%	<b>3.46%</b>
LOG	SGD	5.23%	3.87%	5.42%
	IMPLICIT	<b>4.28%</b>	<b>3.69%</b>	<b>4.01%</b>

Table 7: Summary of data sets and the  $L_2$  regularization parameter  $\lambda$  used

	description	type	features	training set	test set	$\lambda$
covtype	forest cover type	sparse	54	464,809	116,203	$10^{-6}$
delta	synthetic data	dense	500	450,000	50,000	$10^{-2}$
rcv1	text data	sparse	47,152	781,265	23,149	$10^{-5}$
mnist	digit image features	dense	784	60,000	10,000	$10^{-3}$



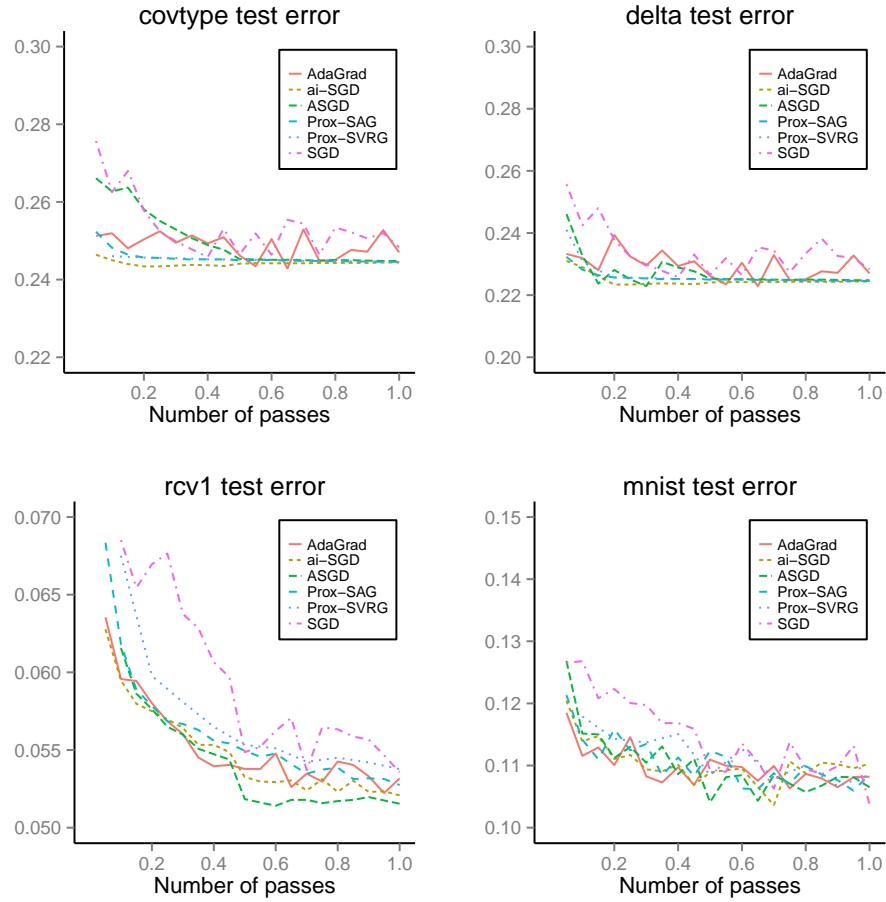


Figure 10: Large scale linear classification with log loss on four data sets. Each plot indicates the test error of various stochastic gradient methods over a single pass of the data.